TABLE OF CONTENTS

Introduction and Prerequisites .......................................................... 1
SDT Layout ....................................................................................... 1
  Reusable Sets Library ................................................................. 1
  Reusable Set Elements .............................................................. 2
  Scenario .................................................................................... 2
  Projects ....................................................................................... 2
Using the SDT .................................................................................. 3
  Building Game Scenarios ............................................................ 3
  Running Game Scenarios ........................................................... 3
  Importing Scenario Definition Files ........................................... 3
Tutorial ......................................................................................... 4
Scenario Development Tool Forms .................................................... 14
  Scenario Form ........................................................................... 14
  Network Form .......................................................................... 16
  Department Form ........................................................................ 16
  Zone Form ................................................................................ 17
  Secrecy Form ............................................................................ 18
  Integrity Form .............................................................................. 19
  DAC Group Form ........................................................................ 20
  Asset Form .................................................................................. 21
  Goal Form .................................................................................. 23
  User Form ................................................................................... 24
  Support Staff Form ..................................................................... 26
  Physical Component Form ........................................................... 27
  Catalogue Component Form: ....................................................... 29
  Filter Form .................................................................................. 30
  Workspace Form ......................................................................... 31
  Condition Form ........................................................................... 32
  Trigger Form ............................................................................... 33
  Procedural Settings Form ............................................................ 35
  Component Network Connection Form ......................................... 36
  Phase Form .................................................................................. 37
  Objective Form .......................................................................... 38
Trigger Tables ..................................................................................... 39
  AttackTrigger ............................................................................ 39
  CashTrigger ............................................................................... 39
  ChangeAssetUsageTrigger ........................................................ 39
  ChangeEnclyoTrigger ................................................................. 39
  ChangeLabelMotive ..................................................................... 40
  ChangeUserDesc ......................................................................... 40
  ClearMalware .............................................................................. 40
  ClearTriggeredThoughts ............................................................. 40
  HappyAdjTrigger ........................................................................ 40
  HelpTipTrigger .......................................................................... 40

http://cisr.nps.navy.mil
LogTrigger ............................................................... 41
LoseTrigger ............................................................. 41
MaskAttackTrigger ....................................................... 41
MessageTrigger ........................................................ 41
PlayMovie ................................................................. 41
ProdAdjTrigger ........................................................... 41
QuestionTrigger .......................................................... 42
QuitGameTrigger ........................................................... 42
ResetAssetAttacked ......................................................... 42
ResetEncyclopedia ........................................................... 42
SetObjectiveStatus ......................................................... 42
SetPhase ................................................................. 42
SetUserThought ............................................................ 42
SpeakTrigger .............................................................. 43
TickerTrigger .............................................................. 43
TrustAdjTrigger ............................................................ 43
UserWhineTrigger ............................................................ 43
WinTrigger ................................................................. 44
Condition Tables .......................................................... 44
AllAssetGoalsMet ............................................................ 44
AssetAttacked .............................................................. 44
AssetComputerHasMAC ................................................ 44
AssetComputerHasPolicy ................................................. 44
AssetInZone ............................................................... 45
AssetZoneHasPolicy ......................................................... 45
AssetToListByFilterType ................................................... 45
AssetToListFilterCount ...................................................... 45
AssignedComputerHas .................................................... 47
AssignedComputerZoneHasPolicy ...................................... 48
AvgUserHappiness .......................................................... 48
AvgUserProd ............................................................... 49
AvgCash ................................................................. 49
CompanyHasComputers ................................................... 49
CompanyHasMACComputers ............................................... 49
Custom ................................................................. 49
GameOnScreen ............................................................ 49
GameStateInfo ............................................................. 50
ItSecStatus ................................................................. 50
MaxCashOnHand ............................................................ 50
MinCashOnHand ............................................................ 50
NumComputersOnNetwork ............................................... 50
ObjectiveCompleted ....................................................... 51
PhaseCompleted ............................................................ 51
Register ................................................................. 51
TimeCondition ............................................................. 51
TriggerGoneOff ............................................................ 51
UserHappiness.................................................................52
UserProductivity..........................................................52
UserTraining .................................................................52
ZoneHasPolicy .............................................................52
UserAccessToAG ...........................................................52
UserHasAssignedComputer .............................................53
UserFailsGoal ...............................................................53
ZoneHasSecurityValue ....................................................53
Attack Types ...............................................................53
Valid O/S – Hardware Combinations ...................................54
Valid Software – O/S Combinations .....................................55
User Thought Priorities .....................................................55
Troubleshooting and Problems ..........................................58
Conditions, Triggers and other Configurable Data ..................60
Introduction and Prerequisites

The Scenario Development Tool (SDT) automates the syntax of the CyberCIEGE scenario definition language through the use of reusable libraries and forms having pull down menus. The basic elements of this language can be reviewed via the SDT forms section of this manual. Knowledge of the CyberCIEGE game is also required. Play some sample scenarios and browse the CyberCIEGE Encyclopedia to learn about the game.

It is suggested that you read through the following sections to become familiar with the basic structure of the tool. Then follow the tutorial to learn some of the mechanics of using the SDT.

SDT Layout

Reusable Sets Library

The upper left window contains folders for each set type (e.g., "User"). Each folder can contain a multiple sets and each set can contain any number of elements. Left clicking on
a folder (e.g., “Asset”) and selecting “new” creates a new library set. Double clicking on a set within a folder will open the set in the tabbed work area. The "File/Save As" menu item creates copies of entire sets.

Reusable Set Elements

When a reusable set is opened, it is added to the tabbed work area. When a set's tab is selected, one element of the set is displayed. Use the "Set Element Management" pull-down list to select the element to display and/or edit. Click the "Add" button to create a new element having initial values copied from the previously displayed set element. New elements are created beneath the previous element, which is useful when defining order-dependent elements such as triggers and phases.

Scenario

The lower left window contains the sets contained in the current scenario. Only one scenario is "open" at a time. The left-most tab in the tabbed work area is always the scenario form. The current scenario can be changed via the "File/Open Scenario" menu. Add a library set to the current scenario by right clicking on the set in the Reusable Sets Library, or by right clicking on the tab if the set is displayed in the tabbed work area. All elements of the selected set are added to the scenario.

Remove element sets from the scenario by locating them in the scenario windowpane and right clicking on them and selecting “remove”.

The "Extra Syntax" field in the scenario form is intended to handle game syntax additions that are not yet designed into the SDT.

Projects

The Reusable Sets Library window pane displays all reusable sets within the current project, regardless of whether they are in the current scenario. A different project can be selected using the Tools/Project Settings menu.

The preferred way to create a new project is to copy an existing project using the “Tools/Clone Project” menu. First select the project you wish to copy, then select “Tools/Clone Project”. In the file dialog, create and select a new directory having the name you wish to use for the project.

If you select (or create) an empty folder, the tool will create all the necessary subdirectories. Two ways of populating an empty project are:

1) use the SDT to create new sets of elements and add them to the scenario;
2) Use the "Tools/Import SDF" menu (see below) to import an existing scenario definition file (SDF) that was created with the "Tools/Build" function in some other project, or saved by the CyberCIEGE game, or was manually created.
Please note that re-usable sets libraries are not implicitly shared between different projects. They are intended to be shared between scenarios within the same project.

**Using the SDT**

**Building Game Scenarios**

Build and test your scenarios incrementally to limit the scope of possible errors. CyberCIEGE error checking and reporting is not very robust, so be prepared to back out changes to get back to a working version of your scenario.

It is strongly recommended that you follow the tutorial section of this document when creating your first scenario.

**Running Game Scenarios**

Before running a game, use the "Tools/Validate" function to check for references to non-existent set elements (e.g., assigning a non-existent user to a component). Use "Tools/Build" to build the SDF file and use "Tools/Run" to execute the result. Running the game requires that the "Tools/Project Settings" Game Directory point to the games "exec" directory. The SDT will copy the SDF into a file called "tmp.sdf" and it will copy the newly created workspace file into the game directory as well.

"View/SDF" displays the SDF built via the "Tools/Build" function using the viewing program named in the "Tools/Project Settings" Viewing program entry. This is initially set to "WordPad".

Note that error checking is very limited. Use the Tools menu to view the game log and the "ParseErrors.txt" file created in the game directory when the game detects errors in the SDF. Finally, view the “crash.txt” file from the “View/crash.txt” menu. If the final entry in that file reflects the current date and time, it may give you a hint as to what kind of error the game encountered.

**Importing Scenario Definition Files**

One way to construct an elaborate scenario is to use the SDT to construct the basic environment, and then use the game itself to buy and configure a variety of components. You can then save that game and import it into the SDT for further refinements. SDF’s can be imported using the "Tools/Import SDF" menu item. Note that this results in the creation of sets that each have one element (e.g., a different set for each user). Note that the SDF import function requires that the appropriate workspace file be in the game directory (as described below). Also, within the SDF, comments delimiting the start and end of zone default component procedural settings must correspond to the syntax saved by the game. This is only an issue for manually created SDFs.

When importing SDF’s, the SDT will attempt to reuse existing library sets. This is useful when experimenting with variations on scenarios saved using the game.
Tutorial

This tutorial walks you through the creation of a CyberCIEGE scenario using the Scenario Development Tool. This tutorial does not define the semantics of the selected values. It is strongly suggested that you have the SDT open as a reference to see the meaning of the fields of a given form (e.g., the fields within the Asset form).

1) Establish a Project
Use Tools / Project Settings to select the “StartOffice” scenario. This is a very simple scenario with one user. Then use Tools / Clone Project to create a copy of this scenario. In the file dialog, create and select a new directory called “MyStartOffice”.

The “MyStartOffice” directory is now where your new project resides. The above step is important because changes you make in this directory will not be over-written by CyberCIEGE updates.

2) Add an Asset
Right click on the Asset folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Joe’s Assets”. Then give the first asset a name, e.g., “Joe’s Spreadsheet”. This results in a display of the asset form. Fill it out as follows:

• Give the new asset a description.
• Add an “Intended Access Control List” entry that permits Joe to read and write the asset (select Joe from the user pull-down list and then click “ADD”).
• Add a “Cost List” entry that reflects a cost of $1000 and a motive of 80 should “Public” be able to read this asset. (enter “1000” in the “Cost” field and “80” in the “Attacker Motive” field and click “ADD”).
• Add the newly created Asset Set to the scenario (right click on the tab and select “Add Tab To Scenario”.

3) Add an Asset Goal
Right click on the Goal folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Joe’s Asset Goals”. Then give the first asset goal a name, e.g., “Modify Joe’s Spreadsheet”. This results in a display of the asset goal form. Fill it out as follows:

• Give the new asset goal a description.
• Add the “Joe’s Spreadsheet” asset to the “Assets Defined for this Goal”. (Select “Joe’s Spreadsheet” and select “Y” for read and write. Then click “ADD”)
• Add the newly created Goal set to the scenario (right click on the tab and select “Add Tab to Scenario”.

4) Give Joe the new goal
Expand the User folder in the Reusable Sets Library pane (if it is not already expanded). Double-click on the set named “Joe”. That should display the user form already
populated with user Joe. Give Joe the goal by adding an entry to the “Asset Goals” table. The “Modify Joe’s Spreadsheet” should be the selected goal. Enter a value of 50 for each of the “Asset Usage”, “Happiness” and “Productivity” fields and then click “ADD”.

5) Run the Scenario
Select Validate/Build/Run from the Tools menu. When the game starts, click the “Play” button. What are Joe’s thoughts? Double click on him. Is he achieving his goal? Why not? Go back to the OFFICE tab. Why is there a penalty?
- Use the buy button to purchase a workstation and place it on Joe’s desk.
- Is Joe now achieving his goal?
- Has the penalty become a bonus?
- Speed up the game to 16x normal speed by pressing the “c” key.
- You may or may not see Joe’s machine crashing and rebooting. Eventually the asset will get compromis

6) Review the log
Selecting “View Log” from the SDT “View” menu. Find the log entry for the asset being attacked. Use the log filter to find “assetAttacked” entries. What happened?

7) Rerun the Scenario and Prevent the Break-in
Double-click on the zone containing Joe’s workspace. Alter the Zone Access list so that only Joe is permitted to enter the zone. And buy a key lock. Notice the resulting “Phys Security” value. Since it now exceeds the motive of the asset, the physical attacks should now fail. Buy the workstation, unpause the game and speed things up. No breakin right?

8) Prevent malware (at least in this scenario) – Procedural Settings & Zone Defaults
You likely noticed the computer was crashing, and maybe you noticed its availability went down. Why was that? Were you able to see the malicious software on the machine?
- Replay the game, preventing physical break-ins per the above. After buying the component, go to the COMPONENT screen and select “No External Software” from the procedural list. Speed up the game. No malware right?
- Run the game again, but this time, before buying the component, go to the zone that will contain the component and set the “No External Software”. This becomes the default for all machines introduced into that zone. Now buy the component. Look at its settings and note how “No External Software” is checked.
- In the SDT, look at the zone form for the zone into which you were placing the computer (expand “Zone” in the scenario elements list and double-click on “3a”. Then select the zone, e.g., “lower left”, from the “Set Element Management” pulldown list.) Find the procedural settings entry. It references a procedural settings set name (to the left of the %) and element of that set (to the right of the %). Find the procedural setting entry. Procedural settings are under the “Other” folder. Expand that and then expand the Procedural Settings folder. Then open the “default” set. Select the “default” element from the “Set Element Management” if it is not already selected. Check the “No External Software”
entry. Now build and run the game. Look at the zone (e.g., lower left”) and notice how the “No External Software” procedural settings is now set.
• You have just reviewed one of the many ways in which the scenario designer can control the initial conditions of the scenario.

9) Add a Web Asset
This step is the first of several that will provide Joe with a goal that can only be achieved if he has an Internet connection.

Right click on the Asset folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Web Assets”. Then give the first asset a name, e.g., “Web Page”. This results in a display of the asset form. Fill it out as follows:
• Give the new asset a description.
• Check the “Instantiated” box – this indicates the scenario will start with this asset already present on a computer (which we will build later).
• Add an “Intended Access Control List” entry that permits Public to read the asset (select Public from the group pull-down list and then click “ADD”).
• Add a “Cost List” entry that reflects a cost of $0 and a motive of 0 should “Public” be able to read this asset. (enter “0” in the “Cost” field and “0” in the “Attacker Motive” field and click “ADD”).
• Add the newly created Asset Set to the scenario (right click on the tab and select “Add Tab To Scenario”)

9) Add an Offsite Physical Component
Put a server in the offsite server rack:
• Expand the Workspace folder in the Scenario Elements pane and open the “WorkspaceStartOffice” set. See the “Workspaces” section of this manual. Then reference the “Offsite Office Grid Layout” figure in this manual. Notice that the server rack in the offsite office is at about grid position 100, 27. Now look at the Workspace table in the form. Scroll down until you find an X-Y entry that is close to 100, 27 and is a server rack (i.e., “S” in the Furn column). That entry says there is a server rack at that X, Y location, which we know is in the offsite office. So we will put the server there. Note the index value (e.g., “16”).
• Right click on the Physical Component folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Offsite Components” and give the first element of the set a name, e.g., “Web Server”. Fill in the form as follows:
  o Enter a description, e.g., “A remote web server.”
  o Enter the position index noted above (e.g., “16”) as the “Grid Position in Office”
  o Find the “Base Component for this Device” and select “Blato Server” from the list”. Select “Populos V9 Server” as its Operating System.
  o Scroll down to “Network Connections”, select “Offsite LAN” and click the >> button.
  o Scroll down to the “Assets”, select “Web Page” and click the >> button.
  o Scroll down to the “Remote User Access List”, select “*.Public” and click the >> button.
Right click on the tab to add the set to the scenario.

- We have made a lot of changes. Make sure things work as expected. In “Tools” select “Validate / Build / Run”. When the game starts, press the “l” key to toggle between the main office and the offsite office. When the offsite office appears confirm there is a computer in the server rack. Select the Components tab and confirm the Web Server contains the Web Page asset.

11) **Add a router to the offsite server rack.**
- Select the “Offsite Components” tab and click the “ADD” button in the Set Element Management bar. Give this second element of the set a name, e.g., “Offsite Router”. (Again, note that sets may have more than one element. There is no semantic difference between sets having one and multiple elements. It is entirely up to the developers preferences for packaging the scenario.) Fill in the form as follows:
  - Enter a description, e.g., “A router.”
  - Enter the position index noted above (e.g., “16”) as the “Grid Position in Office”
  - As the “Base Component for this Device”, select “Bit Flipper” and then “Flip OS” as its Operating System.
  - Scroll down to “Network Connections”, select “Internet” and click the >> button. (Note how it already has “Offsise LAN”. When you “Add” an element to an existing set, it inherits the initial values of the currently selected element.)

10) **Create a new Asset Goal for Joe.**
- Select the “Joe’s Goals” tab. Click “Add” and give the new set element a name, e.g., “Read Web Page”. Give it a description.
- Select the “Joe’s Spreadsheet” goal and click “Remove”. (Note this extra step is a drawback of the feature described above in step 9 under “Network Connections”.
- Select “Web Page” from the list of assets and click the ADD button.
- Then give this goal to Joe as per (4) above. Joe now has two distinct goals.

11) **Validate / Build and run the game.**
Unpause. Check Joe’s goals. Not met right? Buy him a computer. One goal met. Now buy him a router:
- Click the “BUY” button and select the “Network Devices” tab.
- Select the Bit flipper router and click the BUY button. Place the router on Joe’s desk, or perhaps in the on-site server rack.
- Click the “NETWORK” tab. Scroll until you find the router. Select it and then click the “LAN1” and then the “Internet” buttons.
- Scroll until you find Joe’s computer. Select it and click the LAN1 button. You now have a simple network.
- Check Joe’s goals. All met?

12) **Add a simple time condition**
Right click on the Conditions folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Time”. Then give the first condition a name, e.g., “Time1Hour”. This results in a display of the condition form. Fill it out as follows:

- In the “Condition Class”, scroll down and select “TimeCondition”
- Enter a value of “1” for one hour in the “Elapsed Time field.
- Click “Save” and add the condition set to your scenario.

13) Add a simple message trigger

Right click on the Triggers folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Messages”. Then give the first trigger a name, e.g., “One Hour”. This results in a display of the trigger form. Fill it out as follows:

- In the “Trigger Class”, scroll down and select “MessageTrigger”
- Enter “0” for the fixed delay and the random delay.
- Type a message to display to the player in the “Text” field, e.g., “One hour gone…”
- In the “firing condition”, enter your Time1Hour condition (the list of scenario conditions appears in the box on the upper right.)
- Validate build and run. Use the “c” key to speed up time. Your message should pop up after about 1 hour of game time.
- View the log and observe the trigger entry. Note that sometimes you are not sure if triggers have gone off as you intended, and you must check the log.

14) Add a condition to assess asset goals

Right click on the Conditions folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Goals”. Then give the first condition a name, e.g., “JoeFailsSpreadSheet”. This results in a display of the condition form. Fill it out as follows:

- In the “Condition Class”, scroll down and select “UserFailsGoal”
- Select “Joe” as the user, and “Modify Joe’s Spreadsheet” as the goal
- Click “Save” and add the condition set to your scenario.

15) Use a ticker to remind the player to give Joe a computer

Select the “Trigger:Messages” tab and click “NEW” and give the new trigger a name, e.g., “Joe Computer Ticker” This results in a display of a new trigger form. Fill it out as follows:

- In the “Trigger Class”, scroll down and select “TickerTrigger”
- Enter “0.05” for the frequency (the units is days)
- Enter zero for the fixed and random delay.
- Type a message to display to the player in the “Text” field, e.g., “Joe needs a computer…”
- In the “firing condition”, enter: “Time1Hour AND JoeFailsSpreadSheet”
- Validate build and run. Use the “c” key to speed up time. Your message should pop up after about 1 hour of game time. The ticker should start then. And it will repeat about every 0.05 days.
- Buy Joe a computer. The ticker should stop as soon as he achieves the goal.
16) Make Joe speak (Comic book style anyway)
Select the “Trigger:Messages” tab and click “NEW” and give the new trigger a name, e.g., “Joe Computer Thanks” This results in a display of a new trigger form. Fill it out as follows:

- In the “Trigger Class”, scroll down and select “SpeakTrigger”
- Enter 999 as the frequency (this means “once per 999 days”, i.e., just once).
- Enter zero for the fixed and random delay.
- Enter “Joe” as the user field
- Enter the words Joe is to speak in the “Text” field, e.g., “Thanks for the computer!”
- In the “firing condition”, enter: “NOT JoeFailsSpreadSheet”
- Validate build and run. Unpause. Buy Joe a computer. And you thought it was a thankless job.

17) Delay Joe’s Internet Goal
User goals can be hidden from players until a set of conditions are reached. This lets you structure the scenario so the player does not need to do everything at once. Select the User: Joe tab and change his “Read Web Page” goal “Usage” to zero. (Be sure to hit the enter key to make sure the change is recorded.) Then add a trigger to give Joe a non-zero usage of this goal after he has access to his spreadsheet:

- Right click on the Triggers folder in the Reusable Sets Library pane. Select “new” and give the set a name, e.g., “Set Goals”. Then give the first trigger a name, e.g., “Add Joe Internet”. This results in a display of the trigger form.
- In the “Trigger Class” scroll down and select “ChangeAssetUsage”.
- Enter zero for the fixed and random delay.
- Enter “Joe” in the user field
- Enter “Read Web Page” in the goal field (yes, these should eventually be pulldown lists)
- Enter 50 as the new target usage
- In the “firing condition”, enter: “NOT JoeFailsSpreadSheet”
- Add the new trigger set to the scenario
- Validate build and run. Unpause. Check Joe’s goals. He only should have the one. Buy him a computer. Then check his goals. He should now have two goals, one of which (Read Web Page) is not being met.

18) Exert Control Over Attacks
The game engine routinely and randomly attacks assets based on the asset attacker motive and the vulnerabilities resulting from player choices. For many scenarios, the designer desires to make attacks happen more frequently and under selected conditions. Here we will wait about one game hour after Joe gets his computer. We’ll then have outsiders try to breakin to read his spreadsheet.

- Right click on the Triggers folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “Attacks”. Give the first element a name, “Outsider Breakin”.
- In the trigger class, select “AttackTrigger” (even if it is select, you must reselect it to get the parameters to show up.)
• Set the frequency and the fixed delay to 0.05. (about 1 hour)
• In the parameter list, give it a name “Outsider Attack” that will appear in the log.
• Select attack type 18 (see the Attack Types section)
• Provide a motive of “-2”. This instructs the engine to use the motives associated with the assets.
• Set the condition list to “NOT JoeFailsSpreadSheet”.
• Add the set to your scenario.
• Validate / build and run. If the physical security is not altered, the spreadsheet should be compromised an hour (or a bit more) after Joe gets the computer.

19) Remove the “1 Hour” popup message. It is annoying.
• Open the Messages trigger.
• Select “One Hour” in the Set Element Management field, and click the Delete button.

20) Structure your scenario into phases. This is the first of several steps that will structure the existing scenario into two different game phases.
• Right click on the Phases folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “MyPhases”. Give the first element a name, “Phase One”.
• In the form, assign values as follows:
  o Display Name: Work locally
  o Completed Text: Joe can work locally
  o Uncompleted Text: Provide Joe with a safe work environment.
• Click the “Add” button to add a second phase to this set. Name it “Phase Two” and fill in the form as follows:
  o Display Name: Access the Internet
  o Completed Text: [none]
  o Uncompleted Text: Give Joe access to the Internet.
• Add the new Phases set to scenario

21) Define objectives for your first phase. This phase will have two objectives: 1) Joe able to work on the spreadsheet; and 2) Joe working for a while without having the spreadsheet compromised by an attacker.
• Right click on the Objectives folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “Phase1Objectives”. Give the first element a name, “WorkSpreadSheet”.
  o Completed Text: Joe is able to work on his spreadsheet
  o Uncompleted Text: Joe needs a computer with which to work on his spreadsheet.
• Click “Add” to add a second objective. Name it: WorkForAnHour and leave the text fields blank except for Uncompleted Text:
  o Joe must work on the spreadsheet for an hour without having the spreadsheet compromised by an attacker. Check your physical security.
• Note that we don’t need “Completed” text for the last objective since we will move the game to phase 2 as soon as it is achieved. So there is nothing for the player to see.
• Add the Objective set to the scenario. Validate / build and run. Click the “Objectives” button. Note there is only one phase because our second phase currently lacks objectives. We’ll get to those later.

22) Define completeness criteria for the “WorkSpreadSheet” objective.
• Right click on the Triggers folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “Objectives”. Give the first element a name, “WorkSpreadSheet”.
• In the Parameters, set the Objective value to “WorkSpreadSheet” (i.e., the name of the objective in the objective form. Leave the message blank.
• In “OneForMet”, enter a “1”. This sets the objective to “achieved” when the trigger fires.
• In the condition list, enter “NOT JoeFailsSpreadSheet”. Thus, as soon as Joe no longer fails this goal, this objective will be met.

23) Define completeness criteria for the “WorkForAnHour” objective. This objective is a little more subtle. This objective should not be achieved until the player has provided physical security to protect the asset. We could use the “ZoneHasSecurityValue” to measure the security of the zone. However that would not tell us if public has access to the zone. And, the player could choose to secure the entire zone. Or move Joe… In general, it is preferable to let the game engine do the work. So, we will simply measure whether the asset is compromised – or more accurately, not compromised over some period of time. If the asset is compromised, the player can still achieve this objective, but only after the asset is protected and Joe is able to work for an hour or so. We will define a condition to let us determine if the spreadsheet has been compromised; and we will define a trigger that resets this condition after we have measured it so that if the player does fix the problem, we will eventually move to the next phase.
• Right click on the Conditions folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “AssetAttacked”. Give the first element a name, “SpreadSheetAny”. Fill in the form as follows:
  o Select the “AssetAttacked” condition type
  o Select “Joe’s spreadsheet as the asset
  o Enter an “attack type” of -1 to mean “any attack type”
  o Provide a motive range of 0 to 1000.
  o Add the condition set to the scenario
• Open the “Attacks” trigger set. Currently that set contains just one element. Use the Add button and give the new element the name: ResetSpreadSheetAttack and fill in the form as follows:
  o Select ResetAssetAttacked as the trigger type
  o Enter a frequency of 0.05 (about 1 hour)
  o Set the condition to the “SpreadSheetAny” defined above
  o Set the firing condition to just “SpreadSheetAny”
Now note what will happen: The “Breakin” trigger will cause a breakin attack to be attempted. If the physical security is weak, it will succeed causing the “SpreadSheetAny” condition to become true. And this will cause this ResetSpreadSheetAttack trigger to fire right away. The problem is that the condition does not stay true long enough for the game engine to measure it. Remember, we only want to transition to the next phase if this condition stays false for a couple hours or so. Triggers are processed in the order in which they are defined. We’d like to put off resetting the condition until just before the next attack trigger goes off (i.e., until just before it might again be set to true). Thus, we need to reverse the order of the two triggers in this set. This is the purpose of the “Move Up” button. If you click it, it will cause the reset trigger to occur before the attack trigger, which is what we want.

Select the “Trigger: Objectives” tab and click the “Add” button, providing “WorkForAnHour” as the name of the trigger. Fill in the form as follows:

- Select the “SetObjectiveStatus” trigger type
- Provide a fixed delay of 0.1 (about 2 hours) Note the conditions that we define below will have to remain true over the course of two hours for this trigger to go off:
  - Set the Objective to “WorkForAnHour”.
  - Set “OneForMet” to 1.
  - Set the condition to: “NOT SpreadSheetAny AND_NOT JoeFailsSpreadSheet AND Time1Hour”.

24) Define conditions to measure completed objectives.

- Right click on the Conditions folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “Objectives”. Give the first element a name, “WorkSpreadSheet”. Fill in the form as follows:
  - Select ObjectiveCompleted as the condition type
  - Select WorkSpreadSheet as the condition.
- Click Add and provide a condition name of “WorkForAnHour”
  - Select WorkForAnHour as the condition
- Add the new condition set to your scenario

25) Transition to the second phase; and delay Joe’s Internet goal until this phase.

Select the “Trigger: Objectives” form and click “Add”, provide a name of “GoPhase2” and fill in the form as follows:

- Select the “SetPhase” trigger type.
- Enter 999 as the frequency and 0 as the delays.
- Enter “Phase Two” as the name of the phase to set.
- Provide text for a pop-up message, e.g., “Congratulations, you let Joe work safely for a couple hours! Now review your new objectives.”
- Enter: “WorkSpreadSheet AND WorkForAnHour” as the conditions.
- Note this is what was meant earlier when it was suggested that phases only transition based on ObjectiveCompleted conditions. We can now change the meaning of the different objectives without also having to change the conditions under which phases transition.
• Change the “Add Joe Internet” trigger to go off after the transition to phase 2.
  o Select the “Condition: Objectives” tab and click add, naming the new condition: “InPhase2”
  o Select “PhaseCompleted” as the condition class.
  o Select “Phase One” as the phase.
  o Open the “Trigger: Set Goals” set. Change the conditions to read: “InPhase2”.
  o And set the trigger to “runs while paused”. This will reduce the potential delay between phase transition and Joe’s new goal.

26) Define Phase 2 Objectives. Right click on the Objectives folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “Phase2Objectives”. Give the first element a name, “AccessInternet”. Fill in the form as follows:
  • Set the “Phase From” value to 1 (which is the second phase)
  • Set the display name to: Access Internet
  • Set the uncompleted text to “get joe on the internet”
  • Add the set to your scenario.

27) Play the new scenario. Validate / build and run. Confirm that the second phase is not entered until Joe has worked a couple hours safely. When phase 2 is entered, confirm Joe has the new goal.

28) Add win and lose triggers. Open the “Conditions: Goals” set and click “Add” give the new element the name “JoeFailsInternet” and change the Asset Goal to “ReadWebPage”. Click “Save”. Then,

• Right click on the Triggers folder in the Reusable Sets Library pane. Select “new” and give the new set a name, “WinLose”. Give the first element a name, “Win”. Fill in the form as follows:
  o Set the trigger class to “WinTrigger”
  o Set the fixed delay to 0.1 and the random delay to 0
  o Enter text to display on the debriefing window: “Congratulations…”
  o As the conditions, enter: “InPhase2 AND_NOT JoeFailsInternet”
• Open the “Condition: Time” set and click “Add”. Give the new element the name “Time8Hours”. Enter “8” as the elapsed time, and “1” for “OneIfPerPhase”. This means “8 hours have passed in the current phase”.
• Select the “Trigger: WinLose” and click “Add” to add a lose trigger. Give it the name “LoseTimeout”. Fill in the form as follows:
  o Set the trigger class to “LoseTrigger”
  o Set the delays to 0
  o Enter text to display on the debriefing window: “Ooops…”
  o As the conditions, enter: “Time8Hours”
• Add the new set to the scenario
Scenario Development Tool Forms

This section describes the fields of the various SDT forms. Each form defines one of the SDT elements (e.g., users, physical components, goals, etc.). Many form fields define initial values that can then be altered by the player during a game. In these cases, the designer is directed to the CyberCIEGE encyclopedia for descriptions of these fields.

Scenario Form

Organization Name: Not used in the game

Title: Scenario Title appears at the top of the game screen.

Start Up Money: The amount of cash the player starts with.

Budget: Fixed amount of money the player receives per month.

Start Date: Day and time at the start of the scenario

Profit Sharing: Percentage of users salary that goes to player bonus / penalty. The actual profit sharing is also a function of user productivity.

Site Name: Not used in game

Site Description: Not used in game

Office Type: Selects art work for main office and offsite. If there is no offsite, select “none”.

Internet: Whether the scenario has an Internet network (i.e., something available to the public).

Internet Name: The name of the Internet network, defaults to “Internet”.

Static Network: Whether the player can connect/disconnect components from the Internet.

Tutorial Attacks: not used; leave unchecked.

Quit Text: Text displayed when player quits.

Initial Briefing: Displayed on startup screen.

Full Briefing: Displayed in Game screen.
Extra Syntax: Currently used to manage viewpoints, which are a set of camera positions that are cycled through via the “=” key during game play. Intended to provide easy component/user placement. See examples for semantics.

Camera Position: Initial game camera position. See examples for syntax and semantics.

Attack Masks: Attack types the will be masked at the start of the game. See the section on “Attack Types”.

NonServer Default Public: If set, then workstations purchased by players that are not assigned to users will default to local public access.

End on Compromise: Leave unset.

Easy Training: Purchases a lot more training per click.

Easy ACLs: If set, then selecting “Protect with ACL” for a component will cause the ACLs on that component to automatically match the intended access of the asset. If not set, the player will have to manually set the ACLs on the component if “Protect with ACL” is not set.

Use Catalog: Leave set

Networks Everywhere: If set, then if a network exists in more than one zone it is assumed it can be wire-tapped from any zone. Note this is not affected by the zone “excluded networks” setting – though perhaps it should be.

Guards Cost at Startup: If set, hiring a guard will cost one months salary up front.

Attack Tickers: If set, game engine-generated tickers will scroll when assets are attacked.
Network Form

Network Name: The name as it will appear in the NETWORK screen.

Network IP: not used

Static: Whether players can connect/disconnect components on this network.

Department Form

Users can be assigned to a department. There is no semantics to this. Note when defining support staff, the department is either SECURITY or TECH. Do not define these departments.

Name: The name of the department.
Zone Form

Distinct physically securable zones. This includes default procedural settings that will be applied to computer purchased for the zone, and it includes physical security settings. Note users potentially become unhappy with some settings (e.g., cameras). If you set an enterprise user’s “HISupportSkill” to 100, the user will be immune to such things.

Name: The name of the zone as it appears in the scenario. If this is an offsite zone, call it “Offsite”, or put the word “home” or “Home” in it.

Zone Description: Text that appears in the ZONE screen.

Static: Whether the player can alter attributes of this zone, including adding components.

Static Selectable: Like Static (above), but the physical security of the zone is based on selected properties. Otherwise, static zones have ultimate physical security.

Art: name of a .tga file that will be displayed in the ZONE screen when this zone is selected.

Zone Upper Left Corner…: The four corners of the zone in grid coordinates as displayed in the game via the “g” key. Zones are rectangles.

Procedural Settings: Name of procedural setting scenario descriptor. These are defined under the “Other” folder. Per the encyclopedia, these are default values applied to computers that are placed in this zone by the players. Note that once a user has been given procedural security instructions (i.e., been assigned a computer) that users settings will be inherited by any computer subsequently assigned to that user (e.g., if the users computer is stolen and then replaced.)

Guard Faces: If a guard is at the door to this zone, the guard will face in this direction.

Guard X / Y: Location of the guard at the door in grid coordinates per the “g” key.

Constraints on User Access to the Zone: Per the encyclopedia “Zones” entry.

Excluded Networks: A list of networks that cannot be connected to computers that are in this zone.

Configuration Settings: Per the encyclopedia entry for “Components”, these are configuration settings that are inherited by components that are placed in this zone. See the encyclopedia for details.
Secrecy Form

Secrecy Label: A text string to use as a secrecy label tag in this scenario file
For example, “Top Secret”

Level: A single value representing the hierarchical secrecy level between 0 and 64
This value has no semantics other than its relative value to other secrecy levels, i.e., it and the categories below determine the partial ordering.

Category: An enumeration of non-hierarchical secrecy categories, e.g., “A” or “A,B”.
Leave this blank if there are to be no categories.

Value of Secret: how much the player loses if an asset having this label is disclosed to an attacker.

Monthly Change…: not used

Value of Secret for an Attacker: The motive for an attacker to disclose assets having this label. Value is between 0 and 1000.

Initial Background Check: The degree of background checks initially applied to users who have this label as their secrecy clearance.
Integrity Form

Integrity Label: A text string to use as a integrity label tag in this scenario file
For example, “Accounting”

Level: A single value representing the hierarchical integrity level between 0 and 64
This value has no semantics other than its relative value to other integrity levels, i.e., it
and the categories below determine the partial ordering.

Category: An enumeration of non-hierarchical integrity categories, e.g., “A” or “A,B”.
Leave this blank if there are to be no categories.

Value of Integrity: how much the player loses if an asset having this label is modified by
an attacker.

Monthly Change…: not used

Value of Integrity for an Attacker: The motive for an attacker to modify assets having
this label. Value is between 0 and 1000.

Initial Background Check: The degree of background checks initially applied to users
who have this label as their integrity clearance.
DAC Group Form

Users can be assigned to discretionary access control (DAC) groups, which can then be used to define “intended access” of an asset. For example: “all users in the engineering group should be able to read this asset.”

Name: The name of the DAC group, e.g., “engineering”.

Initial Background Check: The degree of background checks initially applied to users who belong to this group.
Asset Form

Asset describes an asset that can exist on one or more components at the start of a scenario. Alternately, assets can be created dynamically by virtual users based on the user goals. In the latter case, asset definitions can include symbolic values that are resolved based on the attributes of the user who creates the asset.

Asset Name: The name as it appears in the game.

Secrecy: The secrecy label of this asset (if any). If secrecy labels are defined, then all assets must have some secrecy label. Disclosure of this asset has a cost and motive derived from the secrecy label in addition to those defined below under the cost list.

Integrity: The integrity label of this asset (if any). If integrity labels are defined, then all assets must have some integrity label. Modification of this asset has a cost and motive derived from the integrity label in addition to those defined below under the cost list.

Denial of Service Motive: Between 0 and 1000, determines the strength of denial of service attacks against this asset.

Availability Penalty: Monthly cost to the player if this asset is not available to users. Since it is monthly, this value must be large to have an effect.

Block User Assign: If set, users cannot re-assign this asset to other components. Only meaningful if “Is Instantiated” is set below.

Create While Paused: If set, users will create this asset (if they can) while the game is paused. Otherwise users do not create assets unless the game is unpaused (i.e., to permit the player to establish the full network and security before users decide where to create assets).

HasDAC: keep this set.

Is Instantiated: If set, this asset must be assigned to a component in a component form. If it is not set, the asset does not exist at the start of the game.

Description: Displayed in the ASSET screen.

Intended Access Control List:

The intended discretionary access to the asset in terms of users or groups of users and the specific access modes. The access modes are: read, write, control and execute. Available modes are Y = Yes, N = No and X = Don't care or "don't care". A entry of:

joe YYYY

means that joe is intended to have read, write, control and execute access to the asset.
The entry:
  joe YNNN
means that joe should be explicitly denied write, control and execute access to the asset.
There may be multiple entries. The most specific entry is applied to each user and their desired mode of access. So, if Joe were within the Engineering group, the two entries:
  joe YNNN
  Engineering YYYY
would still deny Joe write, control and execute access. The "don't care" mode permits a user's access to be whatever is specified for the groups to which the user belongs. For example, if Joe were in the Engineering group, then the two entries:
  joe XNNN
  Engineering YYYY
would mean that Joe should have read access to the asset.
By default, access is denied.

Cost List:

Defines the penalties incurred if a particular access violation occurs based on the "Intended Access Control List" defined above. In such an event, the "Cost List" is referenced to determine the cost based on which one of these groups or people access the asset. Name or group can be expressed as the wildcard character: "**". Mode is read, write, control, execute, or any.

Cost is the cash the player loses in the event of compromise.

Attacker motive is between 0 and 1000.

Mode for the cost list is, from left to right, read/write/control/execute where 'Y' triggers the property while 'N' does not. For example, YYNN would mean the that if read or write access is violated then the penalty amount in the cost field would be triggered; a NNYN would mean that only if the control of the asset is violated would the cost be triggered.
Goal Form

Once defined, a goal that can be given to one or more users. Goals are defined in terms of assets or/and software needed to access assets.

Asset Goal Name: As it appears in the USER screen.

Description: As it appears in the USER screen.

Availability Penalty: Monthly cost to the player if this goal is not achieved by a user who has the goal.

Availability Penalty Change: not used.

Assets Defined for this Goal: If multiple assets are listed within an asset goal then it means that the user has to be able to access those assets at the same time.
  Asset: Name of the asset that must be accessed.
  Filtered: whether user access to the asset could be filtered by a gateway component per the “Filtered Software Type” below. The asset must be on the same component as filtered software.
  Read: Y means the user must be able to read the asset.
  Write: Y means the user must be able to write the asset.
  Execute and Control are not currently used.

Goal is Shared: If set, then if multiple users are assigned this asset goal, they can only succeed in this goal if all users assigned this goal succeed.

Use Other Computers: If set, users will try to use other peoples computers to achieve this goal if they are unable to achieve the goal from their assigned computer.

PromiscDocs: If set, the user must access many external files that might contain macro viruses while achieving this goal.

Software: Specific software applications that must be used when achieving this goal. Note the application can reside on any component to which the user has (potentially remote) access.

Software Type: Specific types software applications that must be used when achieving this goal. Note the application can reside on any component to which the user has (potentially remote) access.

Filtered Software Types: Must also appear in Software Type, filters that sit between the user and the target asset can potentially block the users ability to achieve the goal. Target assets are those marked as “filtered” above in the asset list.
User Form

User Name: As it appears in the game.

Department: one of the departments defined above. This appears in the user description, but has no semantics.

Secrecy: The user’s secrecy clearance (if any). This determines the user’s background checks, which effect the user’s trustworthiness as defined below under “Trust”.

Integrity: The user’s integrity clearance (if any). This determines the user’s background checks, which effect the user’s trustworthiness as defined below under “Trust”.

DAC Groups: The DAC groups to which the user belongs. This potentially effects whether a user has “intended access” to an asset, and whether the user has a cost list motive (see the Asset form “Cost List) to compromise an asset.

Default DAC Group: not used.

Asset Goals: Identifies what assets the user needs to access as defined in the Goals form. Users may have multiple goals. Many users may have the same goal(s).

Asset Goal: name of the goal per the Goals form.

TargetUsage: A target usage of zero is the same as not assigning this user the goal. However, the target usage can be changed to a non-zero value via a trigger, thus introducing a “new” goal for user during the game. Target usage affects users’ ability to share workstations to achieve goals. A workstation used by some user to achieve a goal with a high target usage is not available to other users.

Happiness: A value (integer between 0 and 100) is subtracted from the user’s base happiness if the user fails the goal. Also see the efficiency algorithm described.

Productivity: A value (integer between 0 and 100) is subtracted from the user’s base productivity if the user fails the goal. Productivity can affect the player’s Bonus / Penalty based on the user’s salary and profit sharing as defined in the Scenario Form. Also see the efficiency algorithm described.

Efficiency algorithm:

The location of the component used by the user to achieve the goal can affect the happiness and productivity as follows:

- Working on machine assigned to you = 0% change;
- Working over network = 5% penalty
- Working on any machine in your zone other than the machine assigned to you = 20% penalty;
- Working on any machine not in your zone = 60% penalty.
Trustworthiness: A value between 0 and 100 that contributes to how trustworthy the user is. In the simulation, this value is combined with background checks to compute a value between 0 and 900 as follows:

<table>
<thead>
<tr>
<th>Background Check</th>
<th>User Trust</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Trustworthiness</td>
</tr>
<tr>
<td>Low</td>
<td>50 + Trustworthiness</td>
</tr>
<tr>
<td>Medium</td>
<td>100 + 3 * Trustworthiness</td>
</tr>
<tr>
<td>High</td>
<td>400 + 5 * Trustworthiness</td>
</tr>
</tbody>
</table>

Initial Training: The user’s initial amount of training (between 0 and 100).

Happiness: Initial Happiness level of the user

Productivity: Initial productivity of the user

Skill: A value of 100 will keep the user from becoming unhappy when oppressive zone security measures are taken. Otherwise this value has no meaning.

Grid Position: An integer index into the workspace table. See the “Workspace Form”.

Cost: The user’s monthly salary. This contributes to the bonus / penalty calculation as determined by the “profit sharing” value for the Organization and the user’s productivity.

Gender: male or female
Support Staff Form

Support staff are either technical support (who help keep components available and manage the component configuration settings the player selects) or security, e.g., guards. They can be working at the beginning of a scenario, or they can be made available for players to hire them.

Staff Name: as it appears in the IT Staff screen.

Department: Tech or Security (guards).

Hardware Skills: 0 to 100, contributes to effectiveness of Tech staff

Software Skills: similar to Hardware Skills

People Skills: Required to make other skills more effective, 0 to 100.

Available Date: Blank means working at the start of the scenario. “0” means available for hire immediately. Other values are the number of days into the scenario that the staff becomes available to be hired by the player.

Trustworthiness: Not currently used for support staff.

Initial Training: not used

Happiness: not used

Productivity: not used

Skill: How good a job a guard will do. 0 to 100.

Grid Position in Office: Index into the workspace table defining where the support staff will initially appear. Note that for guards assigned to doors (per the Zone form) they will appear at the location defined in the zone form.

Cost: Monthly cost the player pays for this support staff member. Note that if the Scenario form has “Guards Cost At Startup” selected, then this cost is also paid up front when the staff is hired.

Gender: All support staff are currently male.

Description: As it will appear in the IT Staff screen. Keep it brief.
Physical Component Form

These components exist at the beginning of a scenario.

Component Name: As it appears in the game.

Purchase Price: Not used.

Resale Value: Amount the player receives if the component is scrapped.

Maintenance Cost: not yet used.

Percentage of Time Available: initial component availability. This can decrease due to lack of IT support and malicious software.

Static: If set, the player cannot alter the component.

Grid Position: integer index into the workspace table per the Workspace Form.

HwName: The hardware name displayed when this component is selected. If blank, then name is taken from the base component defined below.

Assigned user: Initial assignment, if any.

Procedural Settings: Initial procedural settings for the component as defined in the encyclopedia entry for “Procedural Security”. This is the name of a descriptor defined in the Others / Procedural folder.

Base Component: Determines the component properties such as whether it is a workstation or a server; whether it is a network device (e.g., router) and its shape in the office (unless it is placed in a server rack). This also determines which operating systems are available for the component.

Operating System: See the encyclopedia entry for Operating Systems.

Software: Initial software installed on the component. See the encyclopedia entry for Applications.

Authentication Server List: List of authentication servers used by this component. The "Component Access List" of each of the authentication servers defines who can remotely access this component.

Network Connections: List of the names of networks as defined in the Network form, or connection names as defined in the “Others / Network Connections” form. The latter is used for connections that have ACLs or labels (i.e., for components having operating systems that enforce a MAC policy).
Assets: Initial set of assets on this component. Note an asset may only be on one component, and each listed asset must have “Is Instantiated” check in the Asset form.

Local Access List: Per the Component Configuration Settings encyclopedia entry.

Remote Access List: Per the Component Configuration Settings encyclopedia entry.

Configuration Settings: Per the Component Configuration Settings encyclopedia entry, except for: Access Controls Fully Configured at Startup. If not set, then the ACLs on assets will be set to public.

Link Encryptor Values: Applies only to components whose “Base Component” (defined above) is a link encryptor.
- Clear Text Network: The network over which clear text is sent and received.
- Encrypted Network: The network over which encrypted text is sent and received.
- Key: The key of each link encryptor on either side of an encrypted network must match for information to properly flow between them. See the encyclopedia entry for Link Encryptors.
Catalogue Component Form:
Note also that components of the same type (e.g., servers) must be grouped together in the SDF.

Component Name: As it appears in the catalogue.

Purchase Price: Cost to player to buy the component.

Resale Value: Amount the player receives if the component is scrapped.

Maintenance Cost: not yet used.

Percentage of Time Available: initial component availability. This can decrease due to lack of IT support and malicious software.

Base Component: Determines the component properties such as whether it is a workstation or a server; whether it is a network device (e.g., router) and its shape in the catalogue and in the office (unless it is placed in a server rack).

Operating System: See the encyclopedia entry for Operating Systems.

Software: Initial software installed on the component. See the encyclopedia entry for Applications.

Configuration Settings: Per the Component Configuration Settings encyclopedia entry, except for: Access Controls Fully Configured at Startup. If not set, then the ACLs on assets will be set to public.
Filter Form

Filters are used within gateway devices to block different kinds of applications from accessing assets through the gateway. This can affect a user's ability to achieve an asset goal, i.e., if the asset goal involves software that is blocked by the filter from reaching the asset. For each software type and network connection to the gateway component, traffic can be blocked in one or both directions. The direction is with respect to the network. For example, "from" refers to traffic from the network to the gateway device. Each filter form only defines one “Filter Blocking Direction”. Thus, if a given gateway is to initially block some applications in one direction, and other applications in other directions, then multiple filter forms for that same component and network connection must be defined.

Filter Name: Just a unique name. It does not appear in the game.

Filter Blocking Direction: To or from the networked named below.

Component Device: the component that contains this filter.

Network: The network to which this filter is applied.

Applications Filtered: The filtered application types. See the Goals form for additional information on how filters can block user goals.
Workspace Form

The workspace form defines which areas of the office can contain desks and server racks. When playing the game, a player can only assign desks, computers, server racks and users to workspaces defined in this form. This form also permits the scenario designer to specify the initial content of a workspace, e.g., a desk or server rack. The columns are defined as follows:

Index
Corresponds to the “Grid Position” values assigned to users and computers in the “User” and “Physical Component” forms.

X and Y
Grid coordinate values that locate a workspace on the floor plan. These coordinate values can be viewed by running the game and pressing the “g” key and panning. Or you can simply refer to the floor plan figures in this document.

Dir
The direction furniture will face. (N, S, E, W).

Furn
The initial furniture as follows: A = desk; S = Server Rack; I = empty.

The refresh button reads the scenario elements and indicates which users and computers are currently assigned to each workspace. This is a developer convenience and it has no effect on the scenario. Note in the case of servers, only one computer and device is listed.
Condition Form

Conditions evaluate game state such as cash-on-hand, time, whether assets have been attacked, etc. Scenario designers can then cause “triggers” (defined in the Triggers Form) to fire based on a Boolean expression of the current game state.

Condition Name: The name that will be used in trigger condition lists. THESE MUST BE SINGLE WORDS WITH NO SPACES;

Condition Class: The type of condition, e.g., UserFailsGoal, PhaseCompleted, etc. See the condition table below.

Parameter List: Values used to assess whether the condition is true or not. Reference the condition table below.
Trigger Form

Triggers permit scenario designers to cause specific game actions to occur in response to a given set of game state conditions. Triggered actions include popup messages, tickers, attacks, loss of money, etc.

Trigger Name: Unique name that will appear in the game log.

Trigger Class: The type of trigger. See the trigger table below for details.

Frequency: The maximum frequency in days that this trigger might fire, e.g., “5” means once per 5 days. And 0.5 means twice a day.

Fixed Delay: A fixed number of days (expressed as a floating point number) after which a triggered event will occur. The triggers conditions must remain true for this duration or the trigger will not fire. Note, if “runs while paused” (see below) is set, this value is real-time seconds.

Random Delay: A random number of days (expressed as a floating point number) that will be added to the Fixed Delay value to determine the period of time over which the trigger’s conditions must be true.

Runs While Paused: If set, the trigger’s conditions will be evaluated even while the game is paused, and the trigger will go off if the conditions, frequency and delays are met. This is useful for help-tip trigger types.

Parameter List: A set of trigger-class dependent values that determine what the game will do when the trigger goes off. E.g., what message to display to the player; how much money to give the player, etc. See the trigger table below for details.

Condition List:

1) You can have up to 16 conditions in a trigger's condition list.
2) They can have up to 16 levels of depth of parenthesis.
3) Conditions must be connected by one of the following logical connectors: "and", "or", "and_not", "or_not"
4) The first condition can have a logical "not" in front of it.
5) Left parenthesis can only be placed directly in front of condition Names (no spaces), right parentheses can only be places directly after condition Names (no spaces).

Here are some examples. All would resolve as then would in a C compiler:

true if and only if ShortTimeCondition is true
ConditionList: ShortTimeCondition :end
true if and only if ShortTimeCondition is false
   not ShortTimeCondition

true if and only if ShortTimeCondition and PlayerBroke are both true
   ShortTimeCondition and PlayerBroke

true if and only if at least one of ShortTimeCondition or PlayerBroke is true
   ShortTimeCondition or PlayerBroke

true if and only if ShortTimeCondition is true and PlayerBroke is false
   ShortTimeCondition and not PlayerBroke

Use of parentheses for nesting are illustrated below:
   not (ShortTimeCondition or PlayerBroke)
   not ShortTimeCondition or (PlayerBroke and KenHappy)
   not (ShortTimeCondition or (PlayerBroke and not KenHappy))

Condition Values: Some conditions have values that can be assessed dynamically. And these values can be provided by the designer as part of the trigger definition rather than as part of the condition definition. For example, you can define a “GameOnScreen” condition called “onscreen” and ignore its value parameter in the condition definition. You can then use this condition within a trigger’s condition list, specifying the desired value in the condition values section. For example, if a condition list read:
   Time1hour and onscreen

And the condition value #2 was set to “2”, then the “onscreen” condition would evaluate to true only if the player was viewing the office screen.
Procedural Settings Form
These elements are referenced by physical components and zones.

Name: The named used within physical component forms and zone forms to reference this procedural setting.

See the procedural security encyclopedia entry for detailed information on the fields within this form.
Component Network Connection Form

These elements are referenced by physical component forms. They define non-trivial network connections, i.e., those that have initial DAC or MAC attributes. Simple network connections can be expressed directly in the physical component form without reference to one of these elements. Currently, the only supported DAC attributes are access control lists that name users or groups of users.

Connection Name: The name used within physical component forms to select this element.

Network: The name of the network that is connected to the component that references this element.

Discretionary Access Controls: See the Component Configuration Settings entry in the encyclopedia for details.

MAC Connection Settings: See the Component Configuration Settings entry in the encyclopedia for details.
Phase Form

Each scenario can be divided into multiple phases, each having objectives the player must achieve prior to moving to the next phase. Phases and objectives are used to structure scenarios to lead players through several steps. The SetPhase trigger is used to transition from one phase to the next based on designer selected conditions. Designers are encouraged to limit these conditions to “ObjectiveCompleted” conditions. The game engine itself does not automatically tie phase transitions to the completion of objectives.

NOTE: Phases are order-dependent. Designers are encouraged to place all phases of a scenario in the same scenario element set, in the order in which the phases are to occur in the game.

PhaseName: As it will appear in logs and the named used in the SetPhase trigger.

Display Name: As it appears at the top the Objective Screen.

Completed Text: Appended to Display name after the phase has completed.

Uncompleted Text: Appended to the Display name until the phase has completed.
Objective Form

Each phase has one or more objectives; and a single objective can exist on multiple phases. The objectives of a given phase are displayed on the Objectives screen. Remaining objectives appear on the right hand side of the screen. Completed objectives appear on the left hand side of the screen. The designer entirely controls whether or not an objective is “complete” using the SetObjectiveStatus trigger type. The ObjectiveCompleted condition can then be used by the designer to assess the current state of any given objective. Designers are encouraged to use ObjectiveCompleted conditions when transitioning between phases.

Objective Name: Used in triggers and conditions.

Phase: identifies the phase(s) that this objective is part of. Integers between 0 (the first phase) and n where n-1 is the number of defined phases in this scenario.

Objective Completed: If selected, the objective is initially “completed”. This can be useful for objectives that are to appear in the middle of a phase based on some player action. To achieve this, leave the “Display Name” and the “Completed Text” blank.

Display Name: The first part of the objective description on either side of the screen. Leave it blank and just use the “Completed Text” and the “Uncompleted Text” to provide a cleaner presentation to the player.

Completed Text: Displayed on the left side of the screen if the objective is achieved.

Uncompleted Text: Displayed on the right side of the screen until the objective is achieved.
### Trigger Tables

#### AttackTrigger
Cause a specific attack to occur.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>AttackIdentifier</td>
<td>Descriptor for log entry.</td>
</tr>
<tr>
<td>Parameter</td>
<td>AttackType</td>
<td>Identifies a specific attack that is to occur. See Attack Types described below.</td>
</tr>
<tr>
<td>Parameter</td>
<td>AttackMotive</td>
<td>The motive of the attack. A value of -2 will not override the motive associated with the assets.</td>
</tr>
</tbody>
</table>

#### CashTrigger
Adjust player’s cash on hand

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Text</td>
<td>Text to display in popup window (if any)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Amount</td>
<td>Dollar value to adjust (+ or -)</td>
</tr>
</tbody>
</table>

#### ChangeAssetUsageTrigger
Adjust an asset goal “TargetUsage” of a given user.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>User</td>
<td>Name of the user whose asset goal is to be adjusted.</td>
</tr>
<tr>
<td>SecondTriggerText</td>
<td>AssetGoal</td>
<td>The name of the asset goal whose TargetUsage is to be adjusted.</td>
</tr>
<tr>
<td>Parameter</td>
<td>TargetUsage</td>
<td>The new target usage to assign to the user’s asset goal.</td>
</tr>
</tbody>
</table>

#### ChangeEnclyoTrigger
Sets the encyclopedia to show the page specified in the trigger the next time the player brings up the encyclopedia.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Target</td>
<td>URL of Specified page of encyclopedia.</td>
</tr>
<tr>
<td>Parameter</td>
<td>OneToLaunch</td>
<td>If 1, launch the encyclopedia.</td>
</tr>
<tr>
<td><strong>ChangeLabelMotive</strong></td>
<td>Change the attacker motive associated with a label</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Parameter #</td>
<td>Name (Displayed in tool)</td>
<td>Description</td>
</tr>
<tr>
<td>TriggerText</td>
<td>Label</td>
<td>The label whose attacker motive is to change</td>
</tr>
<tr>
<td>Parameter</td>
<td>Motive</td>
<td>The new value for the motive</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ChangeUserDesc</strong></th>
<th>Replace the description of a user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter #</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>TriggerText</td>
<td>Name</td>
</tr>
<tr>
<td>SecondTriggerText</td>
<td>NewDescription</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ClearMalware</strong></th>
<th>Remove all malware from all computers</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th><strong>ClearTriggeredThoughts</strong></th>
<th>Clear all triggered thoughts for a given user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter #</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>TriggerText</td>
<td>Name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>HappyAdjTrigger</strong></th>
<th>Add or subtract an amount to the player’s base happiness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter #</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>TriggerText</td>
<td>Text</td>
</tr>
<tr>
<td>Parameter</td>
<td>Amount</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>HelpTipTrigger</strong></th>
<th>Display a help tip balloon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter #</td>
<td>Name (Displayed in Tool)</td>
</tr>
<tr>
<td>TriggerText</td>
<td>Text</td>
</tr>
<tr>
<td>Parameter</td>
<td>Xsize</td>
</tr>
<tr>
<td>Parameter</td>
<td>Ysize</td>
</tr>
<tr>
<td>Parameter</td>
<td>X</td>
</tr>
<tr>
<td>Parameter</td>
<td>Y</td>
</tr>
<tr>
<td>-------------</td>
<td>---</td>
</tr>
<tr>
<td><strong>LogTrigger</strong></td>
<td>Generate a log entry containing specific text.</td>
</tr>
<tr>
<td>Parameter #</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>TriggerText</td>
<td>Text</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Text</td>
<td></td>
</tr>
</tbody>
</table>

| **LoseTrigger** | Results in the player losing the game and the text from the “DebriefLose” being displayed. |
| Parameter # | Name (Displayed in Tool) | Description |
| TriggerText | Text                     |             |

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Text</td>
<td></td>
</tr>
</tbody>
</table>

| **MaskAttackTrigger** | Use to turn masks on or off for all the different attack types. |
| Parameter | Value | See “Attack Types” |
| ValueMask | Set mask to true or false. 1= mask this attack type 0 = don’t mask this attack |

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Text</td>
<td></td>
</tr>
</tbody>
</table>

| **MessageTrigger** | Results in a pop-up message being displayed. |
| Parameter # | Name (Displayed in tool) | Description |
| TriggerText | Text                     |             |

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>file</td>
<td></td>
</tr>
</tbody>
</table>

| **PlayMovie** | Play movie without player intervention |
| Parameter # | Name (Displayed in tool) | Description |
| TriggerText | file                     |             |

| **ProdAdjTrigger** | Add or subtract an amount to the player’s base productivity |
| Parameter # | Name (Displayed in tool) | Description |
| TriggerText | Text                     |             |

| Parameter | Amount | Amount to alter the given user’s productivity. Can be positive or negative. |
### QuestionTrigger
Ask the player a question using a popup

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Text</td>
<td>The text of the question.</td>
</tr>
<tr>
<td>SecondTriggerText</td>
<td>ConditionName</td>
<td>Name of the register condition to modify based on player response. See the “register” condition.</td>
</tr>
</tbody>
</table>

### QuitGameTrigger
Game immediately shuts itself down.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
</table>

### ResetAssetAttacked
Reset an asset attacked condition, intended to be done after a trigger goes off on the condition so the trigger does not go off again unless the asset is attacked again.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Condition Name</td>
<td>The name of the AssetAttacked condition to reset.</td>
</tr>
</tbody>
</table>

### ResetEncyclopedia
Reset the encyclopedia override that was set with the changeEncyloTrigger.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
</table>

### SetObjectiveStatus
Set an objective as being achieved or not

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Objective</td>
<td>Name of the objective</td>
</tr>
<tr>
<td>SecondTriggerText</td>
<td>Message</td>
<td>Optional message to display</td>
</tr>
<tr>
<td>Parameter</td>
<td>OneForMet</td>
<td>1 = met; 0 = not met</td>
</tr>
</tbody>
</table>

### SetPhase
Change the current phase of the game

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>NewPhase</td>
<td>Name of new phase</td>
</tr>
<tr>
<td>SecondTriggerText</td>
<td>Message</td>
<td>Optional text to display</td>
</tr>
</tbody>
</table>

### SetUserThought
Set the thought for a user.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>User</td>
<td>Name of user</td>
</tr>
<tr>
<td>SecondTriggerText</td>
<td>Thought</td>
<td>The user thought</td>
</tr>
<tr>
<td>Parameter</td>
<td>Priority</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The priority of the thought. See the list of User Thought Priorities</td>
</tr>
</tbody>
</table>

**SpeakTrigger**

Like help tips, but balloons come from users.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>User</td>
<td>Name of user</td>
</tr>
<tr>
<td>SecondTriggerText</td>
<td>Words</td>
<td>What the user says</td>
</tr>
<tr>
<td>Parameter</td>
<td>OneToFreeze</td>
<td>Set to 1 if time is to freeze until player clicks mouse.</td>
</tr>
</tbody>
</table>

**TickerTrigger**

Results in a ticker message being queued for display.

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Text</td>
<td>The text displayed to the user when the trigger occurs.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Repeat</td>
<td>The number of times the message is to cycle through the ticker.</td>
</tr>
</tbody>
</table>

**TrustAdjTrigger**

Alter a user's base trustworthiness

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>User</td>
<td>Name of user</td>
</tr>
<tr>
<td>Parameter</td>
<td>TrustChange</td>
<td>New base trustworthiness</td>
</tr>
</tbody>
</table>

**UserWhineTrigger**

Display message if user is unable to achieve goals as follows:
0 – achieve goal entirely assigned computer
1 – achieve goal using network
2 – must use other computer in same zone
3 – must leave zone
4 – no access

<table>
<thead>
<tr>
<th>Parameter #</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>User</td>
<td>The user whose goal achievement is assessed.</td>
</tr>
<tr>
<td>Parameter</td>
<td>zto4</td>
<td>Least access per above</td>
</tr>
<tr>
<td>Parameter</td>
<td>zto4</td>
<td>Most access per above</td>
</tr>
</tbody>
</table>
### WinTrigger

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in Tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TriggerText</td>
<td>Text</td>
<td>Text displayed upon winning.</td>
</tr>
</tbody>
</table>

Results in the player winning the game and the text from the “DebriefWin” being displayed.

### Condition Tables

#### AllAssetGoalsMet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
</table>

All user asset goals are met.

#### AssetAttacked

<table>
<thead>
<tr>
<th>ConditionText</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssetName</td>
<td>AssetName</td>
<td>Name of asset attacked.</td>
</tr>
</tbody>
</table>

Whether a specific asset has been attacked. (Also see the ResetAssetAttacked trigger).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttackType</td>
<td>See “Attack Types”. A value of -1 denotes any attack type.</td>
</tr>
<tr>
<td>MinMotiveValue</td>
<td>Minimum motive for attack.</td>
</tr>
<tr>
<td>MaxMotiveValue</td>
<td>Maximum motive for attack.</td>
</tr>
</tbody>
</table>

#### AssetComputerHasMAC

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
</table>

Whether the computer containing the named asset enforces a MAC policy.

<table>
<thead>
<tr>
<th>ConditionText</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>Asset name</td>
</tr>
</tbody>
</table>

#### AssetComputerHasPolicy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
</table>

Whether the computer containing a given asset has a given policy.

<table>
<thead>
<tr>
<th>ConditionText</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asset</td>
<td>Asset name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SecondConditionText</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolicyOfConfiguration</td>
<td>See AssignedComputerHas</td>
</tr>
</tbody>
</table>
### AssetInZone

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>Asset</td>
<td>Asset name</td>
</tr>
<tr>
<td>SecondConditionText</td>
<td>Zone</td>
<td>Zone name</td>
</tr>
</tbody>
</table>

Whether a given asset is in a given zone.

### AssetZoneHasPolicy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>Asset</td>
<td>Asset name</td>
</tr>
<tr>
<td>SecondConditionText</td>
<td>PolicyOfZone</td>
<td>PermitEscortedVisitors:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Badges:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ModerateIrisScanner:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExpensiveIrisScanner:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>XRayPackages:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SurveillanceCameras:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Re-enforcedWalls:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ModeratePerimeterAlarms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ExpensivePerimeterAlarms:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProhibitPhoneDevices:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ProhibitMedia:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CipherLockOnDoor:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KeyLockOnDoor:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VisualPeopleInspection:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PatrollingGuard:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GuardAtDoor:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Receptionist</td>
</tr>
</tbody>
</table>

Whether the zone containing a given asset has a specific policy in place.

### AssetToNetworkByFilterType

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>Asset</td>
<td>Asset name</td>
</tr>
<tr>
<td>SecondConditionText</td>
<td>Network</td>
<td>Network name</td>
</tr>
<tr>
<td>ThirdConditionText</td>
<td>SoftwareType</td>
<td>Software Type</td>
</tr>
<tr>
<td>Parameter</td>
<td>Mode</td>
<td>0 = from, 1 = to, 2= both</td>
</tr>
</tbody>
</table>

Whether a named asset can be reached via a given network through a software filter.

### AssetToNetworkFilterCount

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>Asset</td>
<td>Asset name</td>
</tr>
<tr>
<td>SecondConditionText</td>
<td>Network</td>
<td>Network name</td>
</tr>
</tbody>
</table>

The minimum number of filtered software types via which the named asset can be reached via a given network.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mode</th>
<th>0 = from, 1 = to, 2= both</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Count¹</td>
<td>Minimum quantity of filtered software types.</td>
</tr>
<tr>
<td><strong>AssignedComputerHas</strong></td>
<td><strong>Tests whether a specified user’s assigned computer has a specific policy or configuration setting.</strong></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>ConditionText</strong></td>
<td><strong>UserName</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SecondConditionText</strong></td>
<td><strong>PolicyOfConfiguration</strong> “;” <strong>Name of a user, can be “*”:</strong> <strong>The name of the computer setting to test against. Must use internal game name.</strong> <strong>ProtectWithACL:</strong> <strong>WriteDownPassword:</strong> <strong>LockorLogoff:</strong> <strong>PasswordLength:Long</strong> <strong>PasswordLength:Medium</strong> <strong>PasswordLength:Short</strong> <strong>PasswordLength:None</strong> <strong>PasswordCharacterSet:Any</strong> <strong>PasswordCharacterSet:Moderate</strong> <strong>PasswordCharacterSet:Complex</strong> <strong>PasswordChangeFrequency:two</strong> <strong>PasswordChangeFrequency:six</strong> <strong>PasswordChangeFrequency:twelve</strong> <strong>PasswordChangeFrequency:never</strong> <strong>NoEmailAttachmentExecute:</strong> <strong>NoExternalSoftware:</strong> <strong>NoUseOfModems:</strong> <strong>NoMediaLeaveZone:</strong> <strong>NoWebMail:</strong> <strong>ApplyPatches:</strong> <strong>LeaveMachinesOn:</strong> <strong>NoPhysicalModifications:</strong> <strong>UserBackup:</strong> <strong>HoldsUserAsset:</strong> <strong>RemoteAuthentication:</strong> <strong>AcceptPKICerts:</strong> <strong>UseOnTimePasswordToken:</strong> <strong>UseBiometrics:</strong> <strong>UseTokenPKICerts:</strong> <strong>ScanEmailAttachments:</strong> <strong>StripEmailAttachments:</strong> <strong>ChangeFrequency:</strong> <strong>PasswordComplexity:</strong> <strong>AutomaticLockLogout:</strong> <strong>SelfAdminister:</strong> <strong>SelfAdministerMAC:</strong></td>
<td></td>
</tr>
</tbody>
</table>
### Administer Software Control

- Block Removable Media:
- Block Local Storage:
- Browser Settings:
  - Loose
  - Normal
  - Strict
- Email Settings:
  - Loose
  - Normal
  - Strict
- Update Patches:
  - None
  - Regular
  - Automatic
- Update Antivirus:
  - None
  - Regular
  - Automatic
  - As Released
- Config Update Antivirus:
  - None
  - Regular
  - Automatic

### Uninterruptible Power

- Offsite Backup:

### Assigned Computer Zone Has Policy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Condition Text</td>
<td>User</td>
<td>User name</td>
</tr>
<tr>
<td>Second Condition Text</td>
<td>Policy Of Zone</td>
<td>See Asset Zone Has Policy</td>
</tr>
</tbody>
</table>

### Avg User Happiness

The current average user happiness of all workers, not including IT and security, has a given relationship to a given value over a given period of time.

<table>
<thead>
<tr>
<th>Parameter (id)</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>relation0forLessThan</td>
<td>0 means less than test, 1 means greater than test.</td>
</tr>
<tr>
<td>Parameter</td>
<td>TestValue</td>
<td>The amount of happiness to test against.</td>
</tr>
<tr>
<td>AvgUserProd</td>
<td>The current average user productivity of all workers, not including IT and security, has a given relationship to a given value over a given period of time</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Parameter (id)</td>
<td>Name (Displayed in tool)</td>
<td>Description</td>
</tr>
<tr>
<td>Parameter</td>
<td>relation0forLessThan</td>
<td>0 means less than test, 1 means greater than test.</td>
</tr>
<tr>
<td>Parameter</td>
<td>TestValue</td>
<td>The amount of productivity to test against.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AvgCash</th>
<th>Enumeration value that defines the condition class of this condition. Average is taken over a period of 14 days.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter (id)</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>Parameter</td>
<td>relation0forLessThan</td>
</tr>
<tr>
<td>Parameter</td>
<td>TestValue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CompanyHasComputers</th>
<th>Whether the quantity of computers in the enterprise falls between two values, inclusive.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter (id)</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Min</td>
</tr>
<tr>
<td>Parameter</td>
<td>Max</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CompanyHasMACComputers</th>
<th>Whether the quantity of MAC enforcing computers in the enterprise falls between two values, inclusive.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter (id)</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Min</td>
</tr>
<tr>
<td>Parameter</td>
<td>Max</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Custom</th>
<th>Whether a specific custom condition has been met. See “Custom Condition” below. Intended only for tutorial mode of the game.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter (id)</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Condition #</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GameOnScreen</th>
<th>Whether a given screen is being displayed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter (id)</td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Screen¹</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>UI_SCREEN_DEBRIEF 1</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_OFFICE 2</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_NETWORK 3</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_COMPONENT4</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_ZONE 5</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_USER 6</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_ASSET 7</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_GAME 8</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_BUY 9</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_SOFTWARE 10</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_ITMANAGE 11</td>
<td></td>
</tr>
<tr>
<td>UI_SCREEN_OBJECTIVE 12</td>
<td></td>
</tr>
</tbody>
</table>

### GameStateInfo

Whether the game is paused

<table>
<thead>
<tr>
<th>Parameter (id)</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OneForPaused</td>
<td></td>
<td>1=paused 0=running</td>
</tr>
</tbody>
</table>

### ItSecStatus

The percentage staffed of IT staff

<table>
<thead>
<tr>
<th>Parameter (id)</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>forsec</td>
<td>0 = Tech support 1 = security</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td></td>
<td>The minimum staffing for the condition to be true.</td>
</tr>
</tbody>
</table>

### MaxCashOnHand

The Maximum money on hand.

<table>
<thead>
<tr>
<th>Parameter (id)</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MaxCashValue</td>
<td></td>
<td>The amount of cash to test against.</td>
</tr>
</tbody>
</table>

### MinCashOnHand

The minimum money on hand

<table>
<thead>
<tr>
<th>Parameter (id)</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TestValue</td>
<td></td>
<td>The amount of cash to test against.</td>
</tr>
</tbody>
</table>

### NumComputersOnNetwork

The quantity of computers on a given network.

<table>
<thead>
<tr>
<th>ConditionText</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Network</td>
<td>Network name</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Min</strong></td>
<td>Minimum number of connected computers to make this true</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Max</strong></td>
<td>Maximum number of connected computers to make this true</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ObjectiveCompleted</strong></th>
<th>Whether a given object has been completed.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td><strong>ConditionText</strong></td>
<td>Objective</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>PhaseCompleted</strong></th>
<th>Whether a given phase has been completed.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td><strong>ConditionText</strong></td>
<td>“Phase Name”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Register</strong></th>
<th>State variable used with question triggers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td><strong>ConditionText</strong></td>
<td><code>name</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TimeCondition</strong></th>
<th>Whether a given amount of time has passed.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter (id)</strong></td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td>ElapsedTime</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td>OneIfPerPhase</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>TriggerGoneOff</strong></th>
<th>The number of times a give trigger has gone off.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td>Name (Displayed in tool)</td>
</tr>
<tr>
<td><strong>ConditionText</strong></td>
<td>TriggerName</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td>“min #”</td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td>“max #”</td>
</tr>
</tbody>
</table>
### UserHappiness

A given user’s happiness has a given value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>UserName</td>
<td>The user name to be tested. Can be the wildcard “*” to denote any user.</td>
</tr>
<tr>
<td>Parameter</td>
<td>relation0forLessThan</td>
<td>0 means less than test, 1 means greater than test.</td>
</tr>
<tr>
<td>Parameter</td>
<td>TestValue</td>
<td>The amount of happiness to be tested against the user’s current happiness using the given relation.</td>
</tr>
</tbody>
</table>

### UserProductivity

A given user’s productivity has a given value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>UserName</td>
<td>The user name to test. Can be the wildcard “*” to denote any user.</td>
</tr>
<tr>
<td>Parameter</td>
<td>relation0forLessThan</td>
<td>0 means less than test, 1 means greater than test.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>The amount of productivity to be tested against the user’s current productivity using the given relation.</td>
</tr>
</tbody>
</table>

### UserTraining

Amount of training for a given user

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>UserName</td>
<td>User</td>
</tr>
<tr>
<td>Parameter</td>
<td>Min</td>
<td>minimum training</td>
</tr>
<tr>
<td>Parameter</td>
<td>Max</td>
<td>maximum training</td>
</tr>
</tbody>
</table>

### ZoneHasPolicy

Whether a given zone has a given policy

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>Zone</td>
<td>name of zone</td>
</tr>
<tr>
<td>SecondConditionText</td>
<td>PolicyOfZone</td>
<td>see AssetZoneHasPolicy</td>
</tr>
</tbody>
</table>

### UserAcessToAG

Whether a user can achieve a given goal, and the hassle involved.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name (Displayed in tool)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConditionText</td>
<td>User</td>
<td>User name</td>
</tr>
<tr>
<td>SecondConditionText</td>
<td>AssetGoal</td>
<td>Goal name</td>
</tr>
</tbody>
</table>
Parameter | 0to4 | minimum access  
| | | 0 = full access  
| | | 1 = via network  
| | | 2 = not my computer, same zone  
| | | 3 = must leave my zone  
| | | 4 = no access  

Parameter | 0to4 | maximum access per above  

**UserHasAssignedComputer**  
Whether the given user has an assigned computer  

| Parameter | Name (Displayed in tool) | Description  
| | |  
| ConditionText | UserName | User  

**UserFailsGoal**  
A given user goal has been not achieved over a given period of time.  

| Parameter | Name (Displayed in tool) | Description  
| | |  
| ConditionText | UserName | Name of a user, can be “*”.  
| SecondConditionText | AssetGoal | Name of the asset goal not achieved.  

**ZoneHasSecurityValue**  
The amount of physical security in a zone.  

| Parameter | Name (Displayed in tool) | Description  
| | |  
| ConditionText | Zone | Zone  
| Parameter | Min | minimum  
| Parater | Max | maximum  

**Attack Types**  
**TRIG_SOCIAL_RANDBREAKIN 0**  
a random (non-Asset targeted) physical breakin by an external attacker followed by a user bribe in the zone broken into to. The attacker motive/skill can be from 50-100.  

**TRIG_SOCIAL_RAND_OFFSITE 1**  
same as index 0 except the social attack on the user takes place offsite so that attacker does not need to break into the zone with the user.  

**TRIG_SOCIAL_TARG_OFFSITE 2**  
same as attack type index 1, except the attacker targets a specific asset and can be high motive.
TRIG_SOCIAL_TARG_BREAKIN 3
same as attack type index 0, except the attacker targets a specific asset and can be high motive.

TRIG_TROJAN_BREAKIN 4 Outsider breaks in and installs trojan on a machine

TRIG_INTERNET_TROJAN 5 Outsider installs trojan on a machine via the internet

TRIG_AMATUER_HACKER 6 Kiddie internet attacks

TRIG_BAD_POLICIES_USERS 7 Naughty users doing things like making copies they shouldn't and installing software they shouldn't because of bad policies

TRIG_TARG_MAL_SOFTWARE 8 Asset targeted malicious software attack
TRIG_WIRE_TAP 9 Wire tap attack
TRIG_TRAP_DOOR 10 Trap door in operating system or application.
TRIG_THIEF_TARGETED 11 Intruder breakin and stealing hardware with targeted asset on it
TRIG_THIEF_TARGETED_ANY 12 Intruder breaking in and stealing any hardware
TRIG_MEDIA_THIEF 13 intruder stealing computers
TRIG_FAILS23 14 equipment failures
TRIG_DOS 15 Denial of Service attack

TRIG_INSIDER_ALLOWED 16 Corrupted user going after assets they should not have access to
TRIG_INSIDER_HACKING 17 Corrupted user going after assets they should not have don't have access to
TRIG_OUTSIDER_BREAKIN 18 Outsider breaking in zones to get assets. The attacker may also use networks to compromise the asset (e.g., break into the office and use a computer that is networked to the target).

TRIG_OUTSIDER_INTERNET 19 outsider going after assets via the internet

TRIG_VIRUSXprp 20 software copying itself to other machines

Valid O/S – Hardware Combinations

The SDT O/S pulldown lists are a function of the selected hardware. This is maintained in the hwOs.ini as well as the CyberCIEGE source code.
Valid Software – O/S Combinations

The SDT software list is a function of the selected O/S. This is maintained in the swOS.ini file as well as the CyberCIEGE source code.

User Thought Priorities

```c
static UserThought_t _thoughts[THOUGHT_MAX_TYPES] = {
    { 0, NULL, NULL, "I like working here.", THOUGHT_HAPPY,
    THOUGHT_HAPPY_WORK, 3 },
    { 0, NULL, _userThoughtCheckWebBrowser, "I like to surf the web.",
    THOUGHT_HAPPY, THOUGHT_HAPPY_WEBSURF, 3 },
    { 0, NULL, NULL, "How did I get this cushy job?", THOUGHT_HAPPY,
    THOUGHT_HAPPY_CUSHYJOB, 3 },
    { 0, NULL, NULL, "It could be worse I guess.", THOUGHT_NEUTRAL,
    THOUGHT_NEUTRAL_BEWORSE, 3 },
    { 0, NULL, NULL, "Yeah, whatever.", THOUGHT_NEUTRAL,
    THOUGHT_NEUTRAL_WHATEVER, 3 },
    { 0, NULL, NULL, "Another day, another dollar.", THOUGHT_NEUTRAL,
    THOUGHT_NEUTRAL.DAYDOLLAR, 3 },
    { 0, NULL, NULL, "I wish we had better computers.", THOUGHT_FLAVOR,
    THOUGHT_FLAVOR_BETTER_MACHINES, 3 },
    { 0, NULL, NULL, "The mouse is my friend.", THOUGHT_FLAVOR,
    THOUGHT_FLAVOR_MOUSE_FRIEND, 3 },
    { 0, NULL, NULL, "I want to go to lunch.", THOUGHT_FLAVOR,
    THOUGHT_FLAVOR_LUNCH, 3 },
    { 0, NULL, NULL, "I wish I were paid more.", THOUGHT_FLAVOR,
    THOUGHT_FLAVOR_PAIDMORE, 3 },
    { 0, NULL, NULL, "Will I be fired?", THOUGHT_FLAVOR,
    THOUGHT_FLAVOR_FEAR_FIRED, 3 },
    { 0, NULL, NULL, "Will I be promoted?", THOUGHT_FLAVOR,
    THOUGHT_FLAVOR_HOPE_PROMOTE, 3 },
    { 0, NULL, _userThoughtCheckComputer, "My machine is down.",
    THOUGHT_SUPPORT, THOUGHT_SUPPORT_MACHINE_DOWN, 15 },
    { 0, NULL, _userThoughtCheckNetwork, "The network is so slow.",
    THOUGHT_SUPPORT, THOUGHT_SUPPORT_NETWORK_SLOW, 15 },
    { 0, NULL, _userThoughtCheckComputer, "My machine keeps rebooting.",
    THOUGHT_SUPPORT, THOUGHT_SUPPORT_MACHINE_REBOOT, 15 },
    { 0, NULL, _userThoughtCheckSoftware, "My software doesn't run.",
    THOUGHT_SUPPORT, THOUGHT_SUPPORT_SOFTWARE_FAIL, 15 },
    { 0, NULL, _userThoughtCheckComputer, "My computer keeps crashing.",
    THOUGHT_SUPPORT, THOUGHT_SUPPORT_MACHINE_CRASH, 15 },
    { 0, NULL, NULL, "I can't meet my #a goal", THOUGHT_ACCESS,
    THOUGHT_ACCESS_ASSET, 100 },
    { 0, NULL, _userThoughtCheckNetwork, "I can't get onto the network.",
    THOUGHT_ACCESS, THOUGHT_ACCESS_NETWORKFAIL, 15 },

```
{ 0, NULL, NULL, "Where is that file?", THOUGHT_ACCESS, THOUGHT_ACCESS_FILELOST, 3 },
{ 0, NULL, NULL, "I need #s for my machine.", THOUGHT_ACCESS, THOUGHT_SUPPORT_WANT_SOFTWARE, 80 },
{ 0, NULL, NULL, "I don't have rights to access the server.", THOUGHT_ACCESS, THOUGHT_ACCESS_RIGHTS, 3 },
{ 0, NULL, NULL, "IT is our enemy.", THOUGHT_ACCESS, THOUGHT_ACCESS_HATE_IT, 3 },
{ 0, NULL, NULL, "I hate working here.", THOUGHT_COMPLAINT, THOUGHT_COMPLAINT_WORK, 3 },
{ 0, NULL, NULL, "I wish I were in Utah", THOUGHT_COMPLAINT, THOUGHT_COMPLAINT_UTAH, 3 },
{ 0, NULL, NULL, "I can't wait to go home.", THOUGHT_COMPLAINT, THOUGHT_COMPLAINT_GO_HOME, 3 },
{ 0, NULL, NULL, "IT is run by fascists.", THOUGHT_COMPLAINT, THOUGHT_COMPLAINT_IT_FASCISTS, 3 },
{ 0, NULL, _userThoughtCheckNetwork, "I wish we had a faster network connection.", THOUGHT_COMPLAINT, THOUGHT_COMPLAINT_NETWORK_SLOW, 3 },
{ 0, NULL, _userThoughtCheckComputer, "I created an asset", THOUGHT_MAKE_ASSET, THOUGHT_MAKE_ASSET_COPY, 3 },
{ 0, NULL, _userThoughtCheckGuard, "I wonder if the guard's gun is loaded.", THOUGHT_POLICY, THOUGHT_POLICY_GUARD, 6 },
{ 0, NULL, _userThoughtCheckGuard, "That guard just walks around all day.", THOUGHT_POLICY, THOUGHT_POLICY_PATROL, 8 },
{ 0, NULL, _userThoughtCheckGuard, "I hate having security looking at me.", THOUGHT_POLICY, THOUGHT_POLICY_VISUAL, 8 },
{ 0, NULL, _userThoughtCheckPhones, "I really hate not having my cell phone here.", THOUGHT_POLICY, THOUGHT_POLICY_PHONE, 9 },
{ 0, NULL, _userThoughtCheckCamera, "I'm gonna smash that stupid camera one day.", THOUGHT_POLICY, THOUGHT_POLICY_CAMERA, 9 },
{ 0, NULL, _userThoughtCheckXRay, "I really hate having my stuff X-Rayed.", THOUGHT_POLICY, THOUGHT_POLICY_XRAY, 9 },
{ 0, NULL, NULL, "I think Iris Scanners are annoying.", THOUGHT_POLICY, THOUGHT_POLICY_IRIS1, 11 },
{ 0, NULL, _userThoughtCheckBadge, "A badge, yet another nuisance.", THOUGHT_POLICY, THOUGHT_POLICY_BADGE, 7 },
{ 0, NULL, NULL, "I think IT should do backups, not me.", THOUGHT_POLICY, THOUGHT_POLICY_BACKUP1, 5 },
{ 0, NULL, NULL, "I don't like machines without floppys.", THOUGHT_POLICY, THOUGHT_POLICY_MEDIA1, 5 },
{ 0, NULL, _userThoughtCheckConfigMngmt, "Config Management, blah.", THOUGHT_POLICY, THOUGHT_POLICY_CM1, 5 },
{ 0, NULL, _userThoughtCheckPatches, "I don't like apply patches.", THOUGHT_POLICY, THOUGHT_POLICY_PATCHES, 5 },
{ 0, NULL, _userThoughtCheckReceptionist, "I hate that receptionist.", 
THOUGHT_POLICY, THOUGHT_POLICY_RECEPTIONIST_HATE, 5 },
{ 0, NULL, _userThoughtCheckReceptionist, "I like that receptionist.", 
THOUGHT_POLICY, THOUGHT_POLICY_RECEPTIONIST_LIKE, 5 },
{ 0, NULL, _userThoughtCheckEmailStrip, "I don't like my email attachments stripped.", THOUGHT_POLICY, THOUGHT_POLICY_EMAIL_STRIP, 5 },
{ 0, NULL, _userThoughtCheckEmailAttachExecute, "Why can't I execute my email attachment?", THOUGHT_POLICY, 
THOUGHT_POLICY_EMAIL_ATTACH_EXECUTE, 5 },
{ 0, NULL, _userThoughtCheckWebEmail, "Why can't I read my web email account?", THOUGHT_POLICY, THOUGHT_POLICY_WEB_EMAIL, 5 },
{ 0, NULL, NULL, "I don't like #p", THOUGHT_POLICY, 
THOUGHT_POLICY_GENERIC, 2 },
{ 0, NULL, NULL, "Someone stole my computer!!!", THOUGHT_ATTACK, 
THOUGHT_ATTACK_STEALCOMP, 50 },
{ 0, NULL, NULL, "Someone stole my device!!", THOUGHT_ATTACK, 
THOUGHT_ATTACK_STEALDEV, 45 },
{ 0, NULL, NULL, "Uh oh, was it me?", THOUGHT_ATTACK, 
THOUGHT_ATTACK_STEALMEDIA, 30 },
{ 0, NULL, NULL, "Why would IT ask me for my password?", 
THOUGHT_ATTACK, THOUGHT_ATTACK_SOCIAL, 30 },
{ 0, NULL, NULL, "Someone's calling to ask for my password?", 
THOUGHT_ATTACK, THOUGHT_ATTACK_SOCIAL2, 30 },
{ 0, NULL, NULL, "This has got to be a scam...", THOUGHT_ATTACK, 
THOUGHT_ATTACK_SOCIAL3, 30 },
{ 0, NULL, NULL, "I wonder why they asked for this data...", 
THOUGHT_ATTACK, THOUGHT_ATTACK_SOCIAL4, 30 },
{ 0, NULL, NULL, "Mmm, my chair is warm.", THOUGHT_ATTACK, 
THOUGHT_ATTACK_WARMCHAIR, 10 },
{ 0, NULL, NULL, "Should I have done that?", THOUGHT_ATTACK, 
THOUGHT_ATTACK_SHOULD1, 20 },
{ 0, NULL, NULL, "I've got a Virus!", THOUGHT_ATTACK, 
THOUGHT_ATTACK_HAVEVIRUS, 100 },
{ 0, NULL, NULL, "I've discovered a Virus Attempt!", THOUGHT_ATTACK, 
THOUGHT_ATTACK_FOUNDVIRUS, 100 },
{ 0, NULL, NULL, "Please install our special software? OK.", 
THOUGHT_ATTACK, THOUGHT_INSTALL_TROJAN, 30 },
{ 0, NULL, NULL, "Please install our special software? No!", 
THOUGHT_ATTACK, THOUGHT_NOINSTALL_TROJAN, 29 },
{ 0, NULL, NULL, "I think I'll install #s.", THOUGHT_ATTACK, 
THOUGHT_INSTALL_UNAUTHORIZED, 10 },
};
Troubleshooting and Problems

Catalog Components
When you define catalog components in your sdt, take care to group all workstations together; all servers together; and all network devices together. Otherwise the game gets very confused and will not display all your catalog items.

Since the SDT lacks an easy way to move around elements within a set, use a text editor if items get out of order.

If you define multiple sets, the order still matters. Here you can arrange the order by removing all sets from the scenario and then adding them in the desired order.

Triggers
Trigger order matters. If multiple triggers go off at the same time, the engine generally selects the first one in the SDT.

Don’t use colons “:” in descriptions.

Don’t include spaces in user names.
Figure 2: Main Office Grid Layout
Conditions, Triggers and other Configurable Data

The conditions.ini and triggers.ini files allow the addition of new conditions and triggers without code changes. Reference the headers of those files for a description of the syntax.