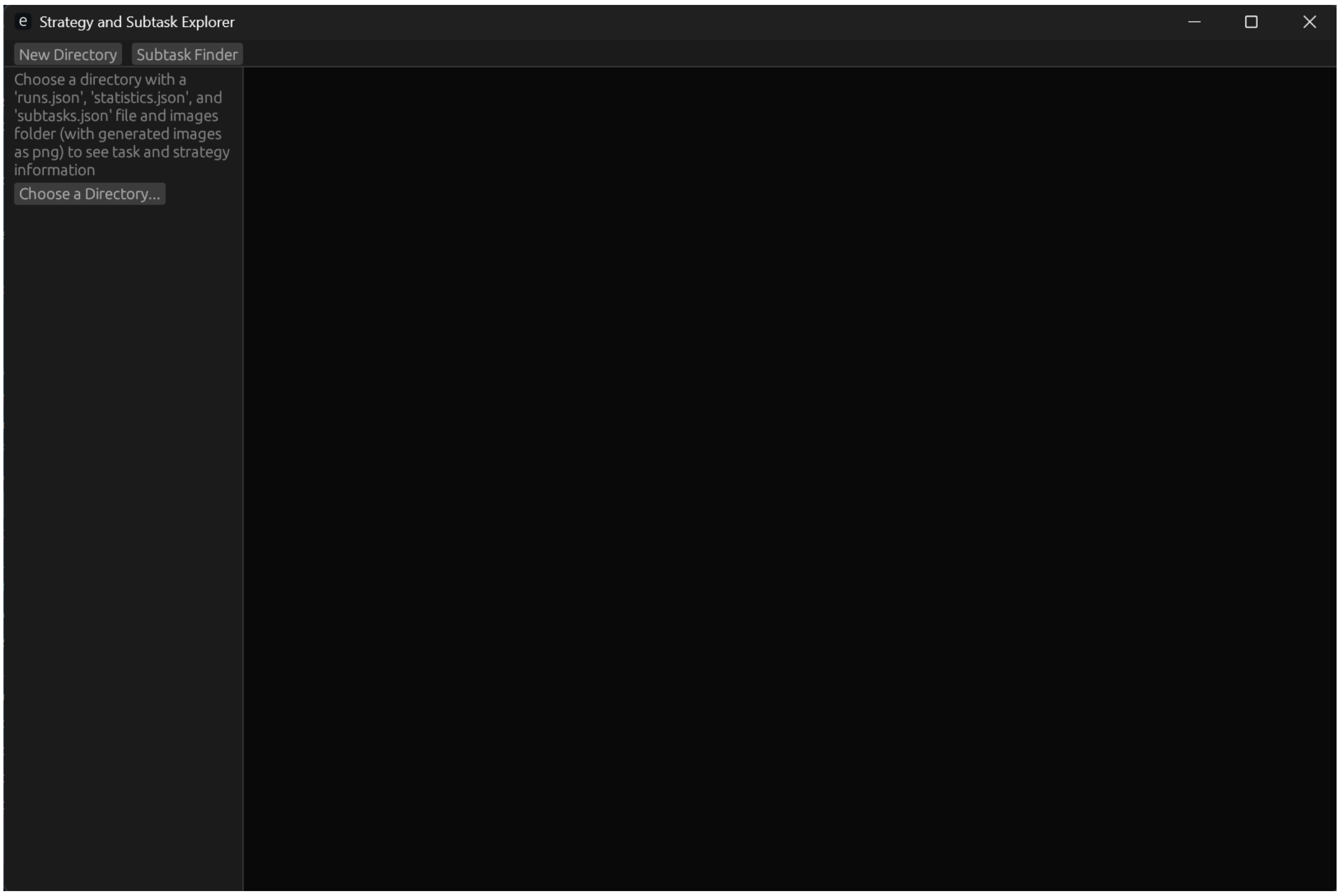
1 Task Explorer GUI

The Task Explorer application is a GUI application built with Rust’s immediate- mode GUI library egui. The Task Explorer application was compiled into a standalone Windows executable that can be run on any modern Windows operating system. The Task Explorer application allows users to easily visualize the results and statistics produced by the task explorer pipeline.

1.1 Data Input/Selecting a New Directory

Upon opening the Task Explorer application, the user will have to select a new directory by selecting the ‘‘New Directory’’ tool and then clicking ‘‘Choose a Directory’’ (Figure 1). The user can then navigate to and select a folder that contains thestatistics.json,runs.json, andsubtask.json artifact files, along with an images folder that contains all artifact images. These artifacts are generated from a Python script that runs the task explorer pipeline on a set of tasks that are completed within an event. A new directory can be chosen at any time to load new results.

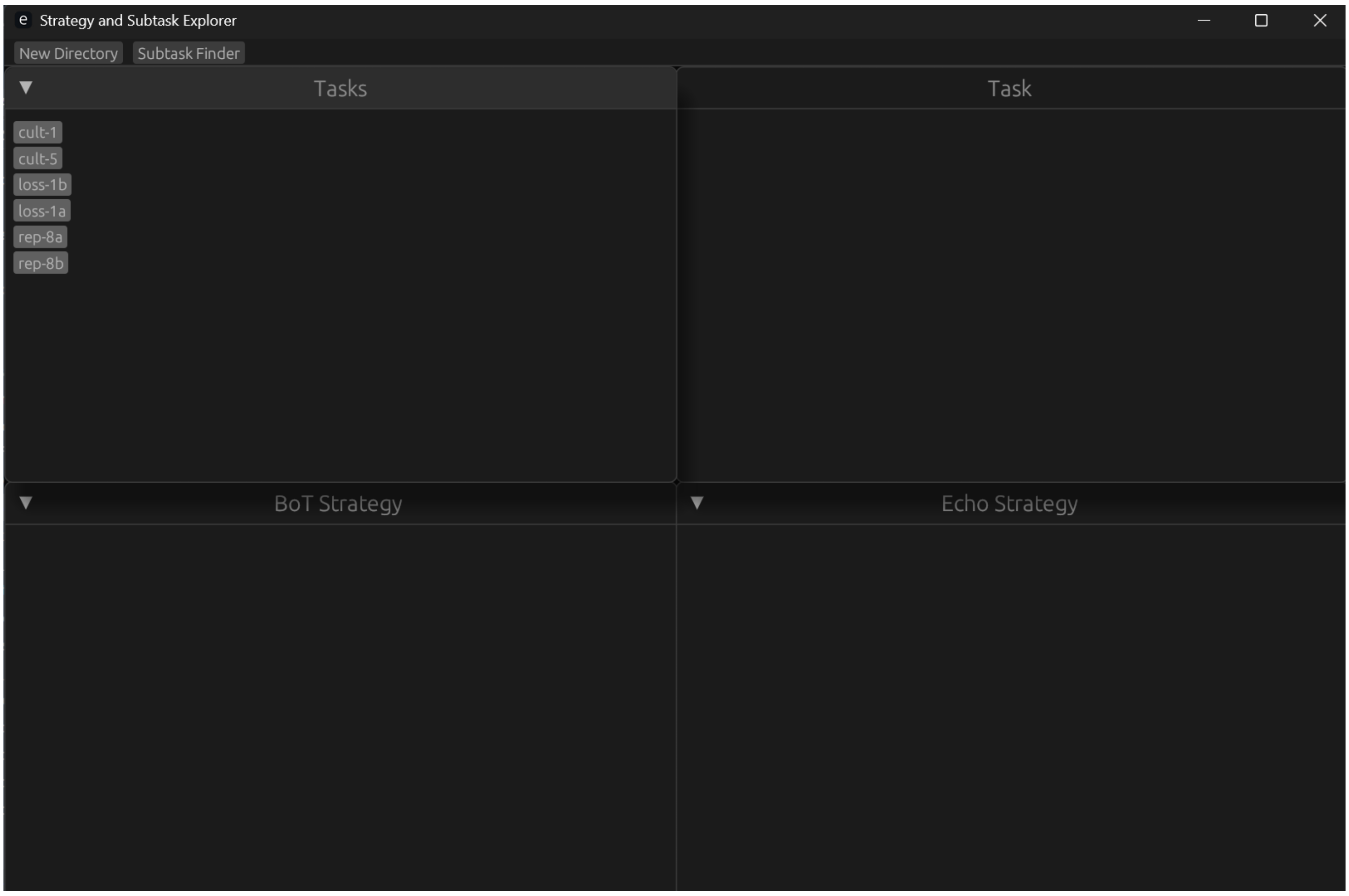


Choose a directory with a 'runs.json', 'statistics.json', and 'subtasks.json' file and images folder (with generated images as png) to see task and strategy information

```
Figure 1: Selecting A New Directory In The Task Explorer Application
```

1.2 The Four Quadrants

After reading the artifact data, the Task Explorer presents four quadrants (Figure 2): a Tasks panel, a Task panel, a BoT Strategy panel, and an Echo Strategy panel. These panels are meant to guide the user through the hierarchical nature of the task explorer pipeline’s results. Users start in the Tasks panel, then move to the Task panel, then to BoT Strategy, and finally to Echo Strategy.



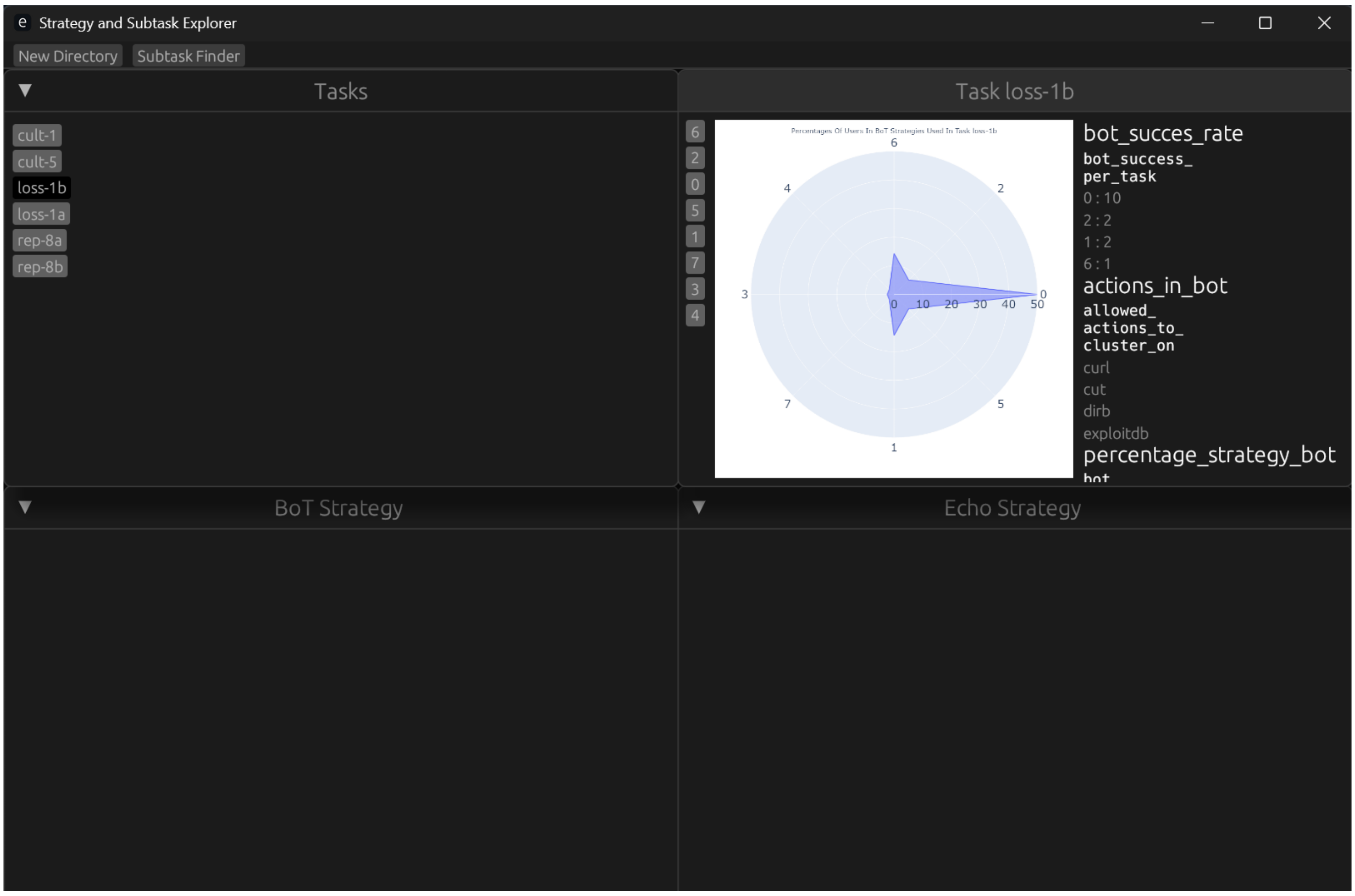
```
Figure 2: The Four Quadrants And Tasks Panel
```

1.3 Tasks Panel

The Tasks panel (Figure 2) displays all tasks that the task explorer pipeline ran on in a given event. Users can click on a task to select it, which updates the Task panel with the selected task's results.

1.4 Task Panel

The Task panel (Figure 3) displays the BoT strategies that were identified within the selected task. Next to the BoT strategies, there is a spider graph to illustrate the percentage of users in each BoT strategy. If you scroll down on the image, the task's hierarchical clustering dendrogram will be just below the spider graph. To the right of these images is a scrollable area that presents any statistics that were computed for the selected task. Users can click on a BoT strategy to select it, which updates the BoT Strategy panel with the selected BoT Strategy's information.



bot_succes_rate

bot_success_
per_task

0 : 10
2 : 2
1 : 2
6 : 1

actions_in_bot

allowed_
actions_to_
cluster_on

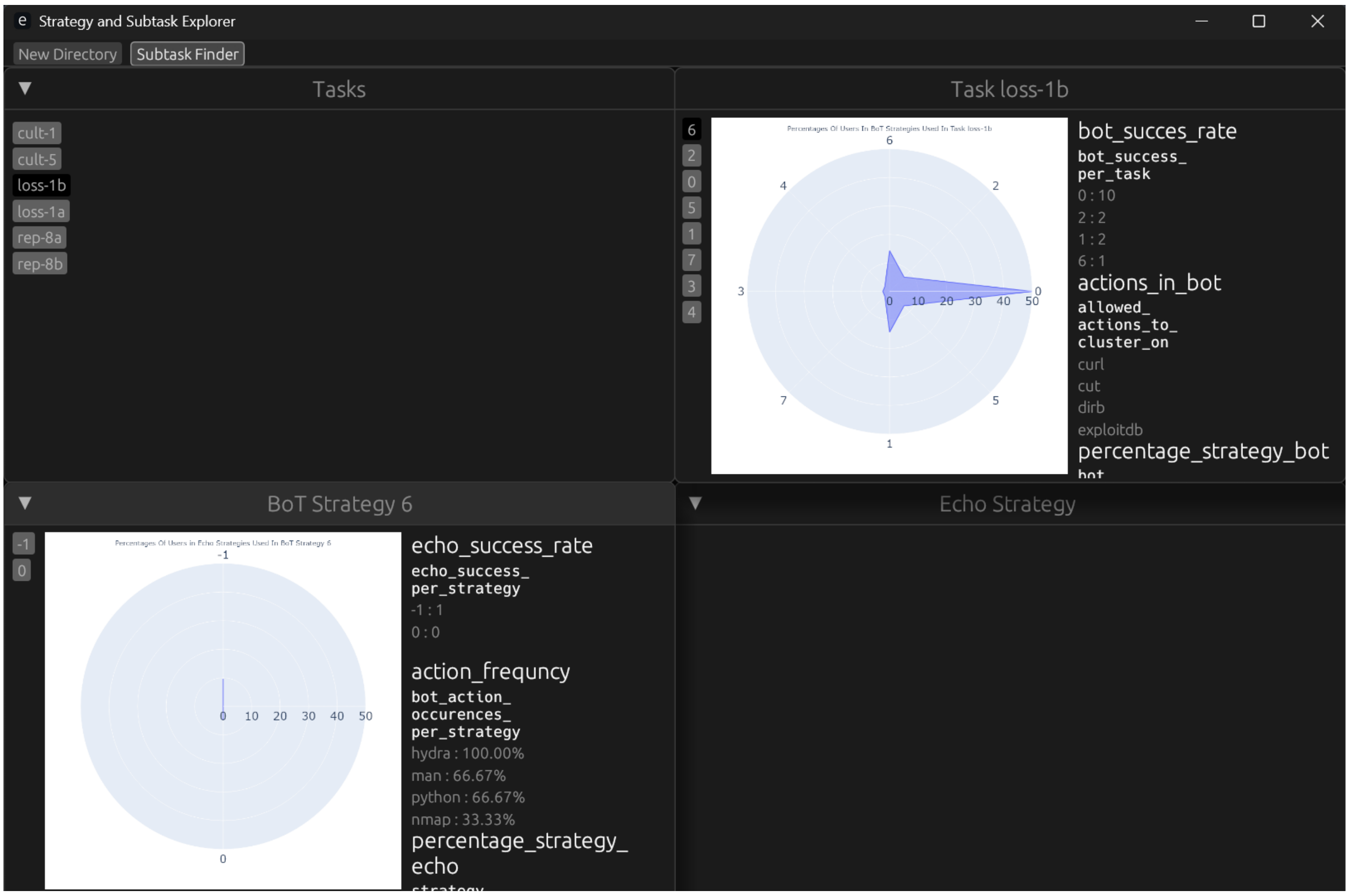
curl
cut
dirb
exploitdb

percentage_strategy_bot

```
Figure 3: The Task Panel
```

1.5 BoT Strategy Panel

The BoT Strategy panel (Figure 4) displays the Echo strategies that were identified within the selected BoT strategy. Next to the Echo strategies, there is a spider graph to illustrate the percentage of users in each Echo strategy. To the right of the graph is a scrollable area that presents any statistics that were computed for the selected BoT strategy. Users can click on an echo strategy to select it, which updates the Echo Strategy panel with the selected echo strategy's information.


Strategy and Subtask Explorer
New Directory
Subtask Finder
Percentages Of Users In BoT Strategies Used In Task loss-1b
Percentages Of Users In Echo Strategies Used In BoT Strategy 6

Tasks

cult-1
cult-5
loss-1b
loss-1a
rep-8a
rep-8b

Task loss-1b

6
2
0
5
1
7
3
4

bot_succes_rate
bot_success_
per_task
0 : 10
2 : 2
1 : 2
6 : 1
actions_in_bot
allowed_
actions_to_
cluster_on
curl
cut
dirb
exploitdb
percentage_strategy_bot
bot

BoT Strategy 6

-1
0

echo_success_rate
echo_success_
per_strategy
-1 : 1
0 : 0

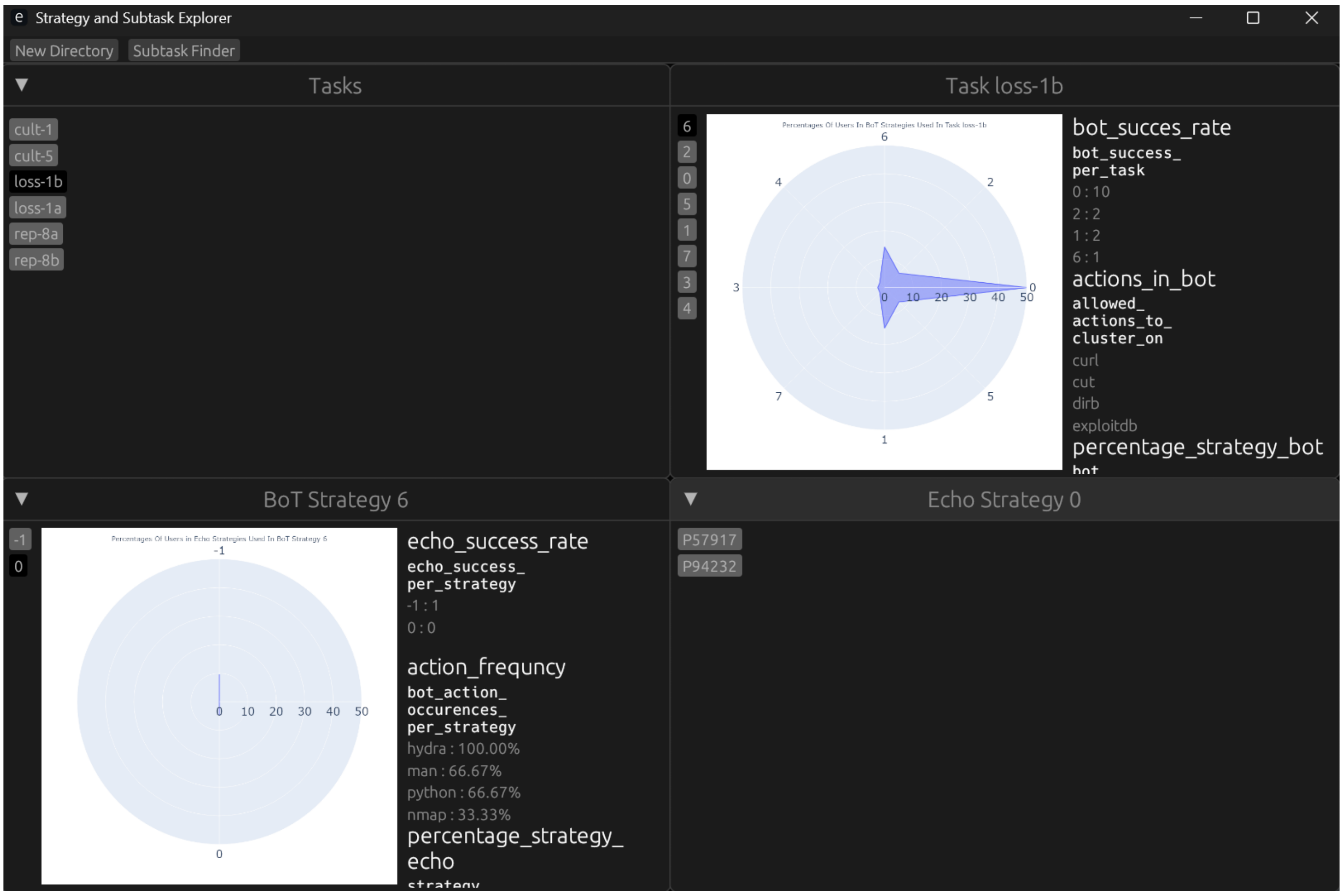
action_frequncy
bot_action_
occurences_
per_strategy
hydra : 100.00%
man : 66.67%
python : 66.67%
nmap : 33.33%
percentage_strategy_
echo
strategy

Echo Strategy

```
Figure 4: The BoT Strategy Panel
```

1.6 Echo Strategy Panel

The Echo Strategy panel (Figure 5) displays the user runs that were grouped within the selected echo strategy. Next to the user runs, any statistics that were computed for the selected echo strategy would show, but currently there are none. Users can click on a run to open the User pop-up for the selected user.


Strategy and Subtask Explorer
New Directory
Subtask Finder
Tasks
Task loss-1b
Percentages Of Users In BoT Strategies Used In Task loss-1b
BoT Strategy 6
Percentages Of Users in Echo Strategies Used In BoT Strategy 6
Echo Strategy 0

cult-1
cult-5
loss-1b
loss-1a
rep-8a
rep-8b

6
2
0
5
1
7
3
4

bot_succes_rate

bot_success_
per_task

0 : 10
2 : 2
1 : 2
6 : 1

actions_in_bot

allowed_
actions_to_
cluster_on

curl
cut
dirb
exploitdb

percentage_strategy_bot

bot

-1
0

echo_success_rate

echo_success_
per_strategy

-1 : 1
0 : 0

action_frequncy

bot_action_
occurences_
per_strategy

hydra : 100.00%
man : 66.67%
python : 66.67%
nmap : 33.33%

percentage_strategy_
echo

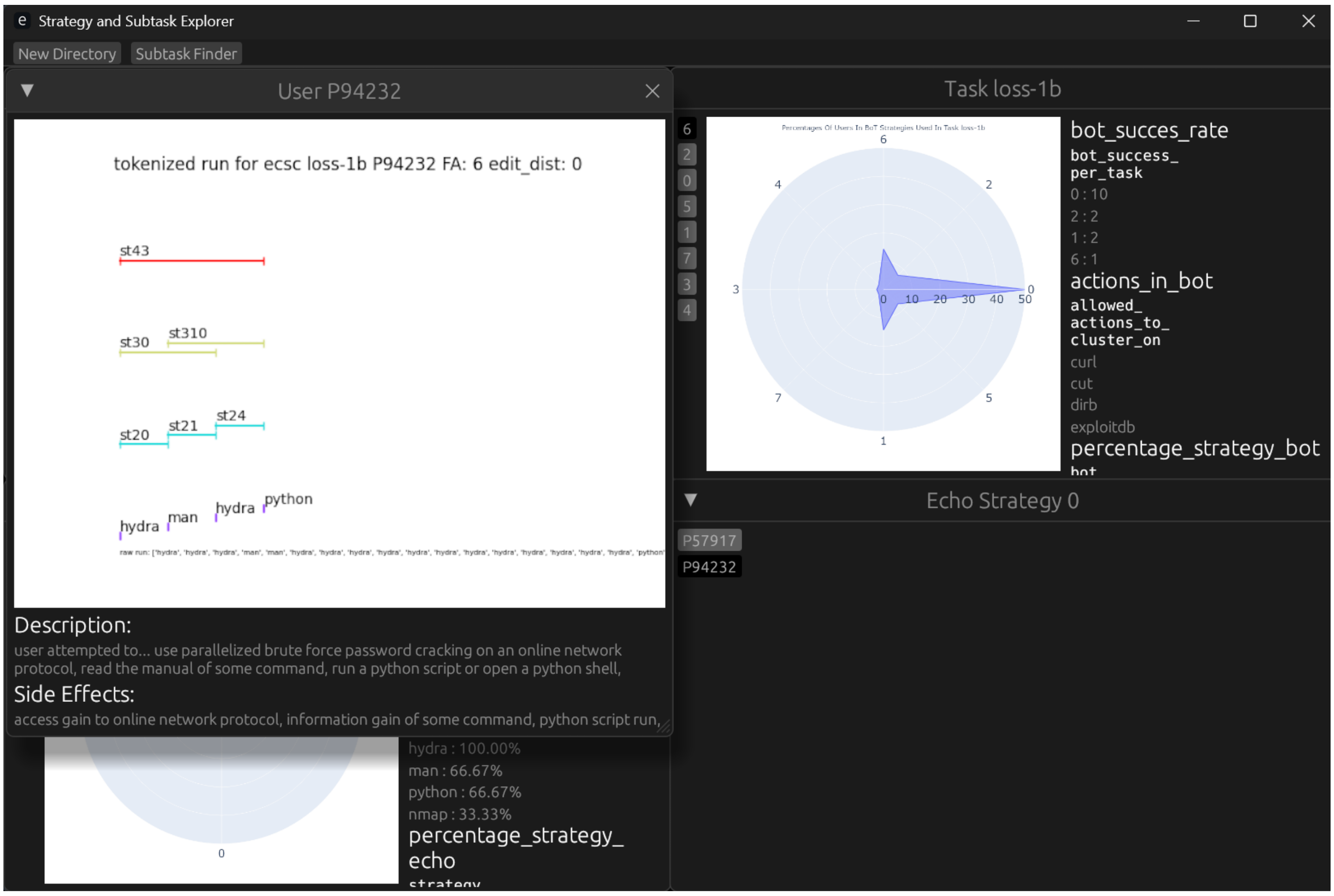
strategy

P57917
P94232

```
Figure 5: The Echo Strategy Panel
```

1.7 User Pop-Up

The User pop-up (Figure 6) displays the selected user’s hierarchical encoding diagram. Under the hierarchical encoding diagram, the run’s description and side effects are listed.



tokenized run for ecsc loss-1b P94232 FA: 6 edit_dist: 0

raw run: ['hydra', 'hydra', 'hydra', 'man', 'man', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'hydra', 'python'

Description:
user attempted to... use parallelized brute force password cracking on an online network protocol, read the manual of some command, run a python script or open a python shell,

Side Effects:
access gain to online network protocol, information gain of some command, python script run,

bot_succes_rate
bot_success_per_task
0 : 10
2 : 2
1 : 2
6 : 1

actions_in_bot
allowed_actions_to_cluster_on
curl
cut
dirb
exploitdb

percentage_strategy_bot
bot

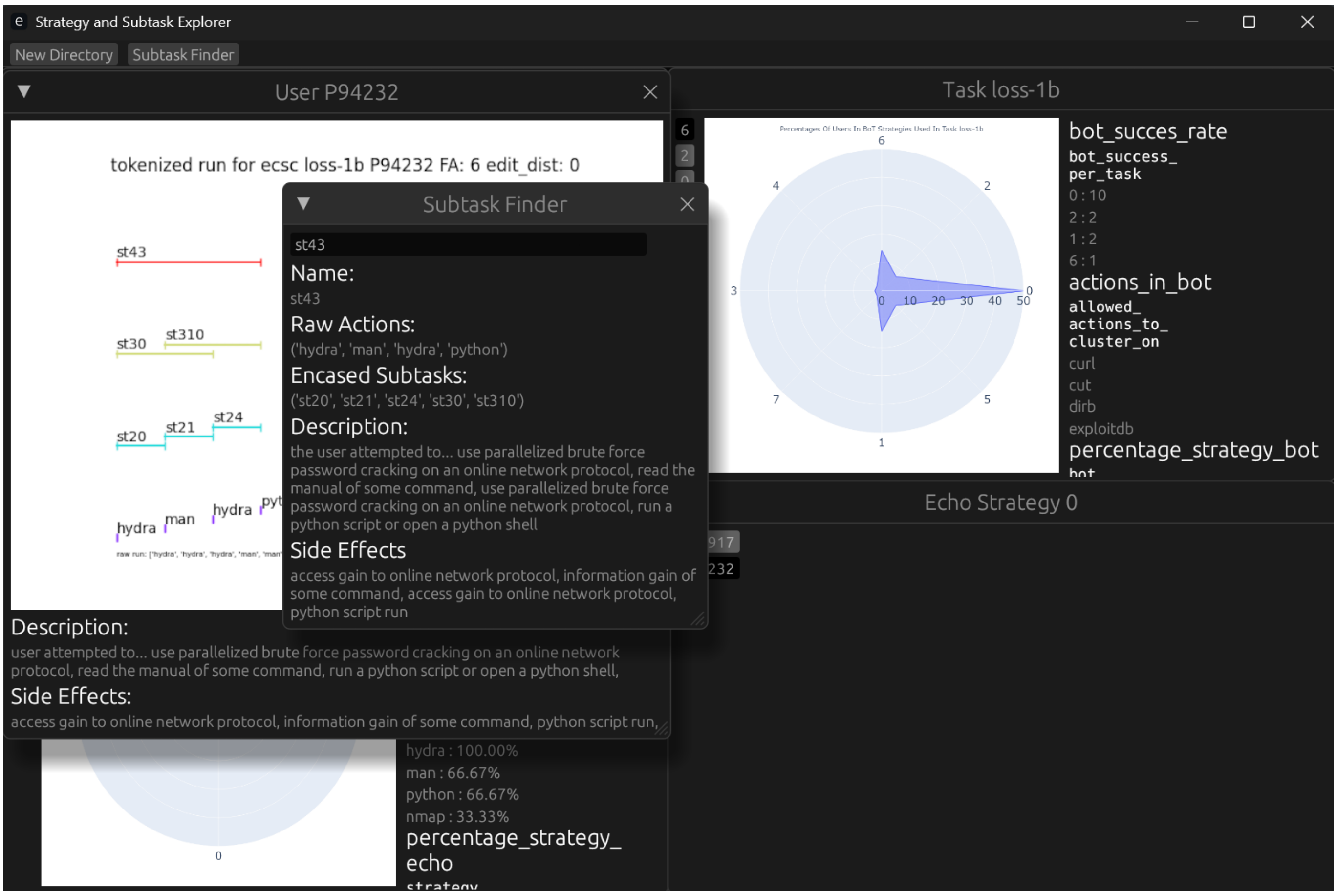
hydra : 100.00%
man : 66.67%
python : 66.67%
nmap : 33.33%

percentage_strategy_echo
strategy

```
Figure 6: The User Pop-Up
```

1.8 Subtask Finder Pop-Up

The Subtask Finder pop-up (Figure 7) can be summoned by clicking on the ‘‘Subtask Finder’’ tool at the top of the application. It allows the user to enter a search for any real subtask and get that subtask’s name, raw actions, encased subtasks, description, and side effects.


Strategy and Subtask Explorer
New Directory
Subtask Finder
User P94232
tokenized run for ecsc loss-1b P94232 FA: 6 edit_dist: 0
st43
st30
st310
st20
st21
st24
hydra
man
hydra
pyt
raw run: ['hydra', 'hydra', 'hydra', 'man', 'man'
Task loss-1b
Percentages Of Users In BoT Strategies Used In Task loss-1b
6
2
0
5
1
7
3
4
0
10
20
30
40
50
Echo Strategy 0
917
232
0

Subtask Finder

st43

Name:

st43

Raw Actions:

('hydra', 'man', 'hydra', 'python')

Encased Subtasks:

('st20', 'st21', 'st24', 'st30', 'st310')

Description:

the user attempted to... use parallelized brute force password cracking on an online network protocol, read the manual of some command, use parallelized brute force password cracking on an online network protocol, run a python script or open a python shell

Side Effects

access gain to online network protocol, information gain of some command, access gain to online network protocol, python script run

Description:

user attempted to... use parallelized brute force password cracking on an online network protocol, read the manual of some command, run a python script or open a python shell,

Side Effects:

access gain to online network protocol, information gain of some command, python script run,

bot_succes_rate

bot_success_
per_task

0 : 10

2 : 2

1 : 2

6 : 1

actions_in_bot

allowed_
actions_to_
cluster_on

curl

cut

dirb

exploitdb

percentage_strategy_bot

bot

hydra : 100.00%

man : 66.67%

python : 66.67%

nmap : 33.33%

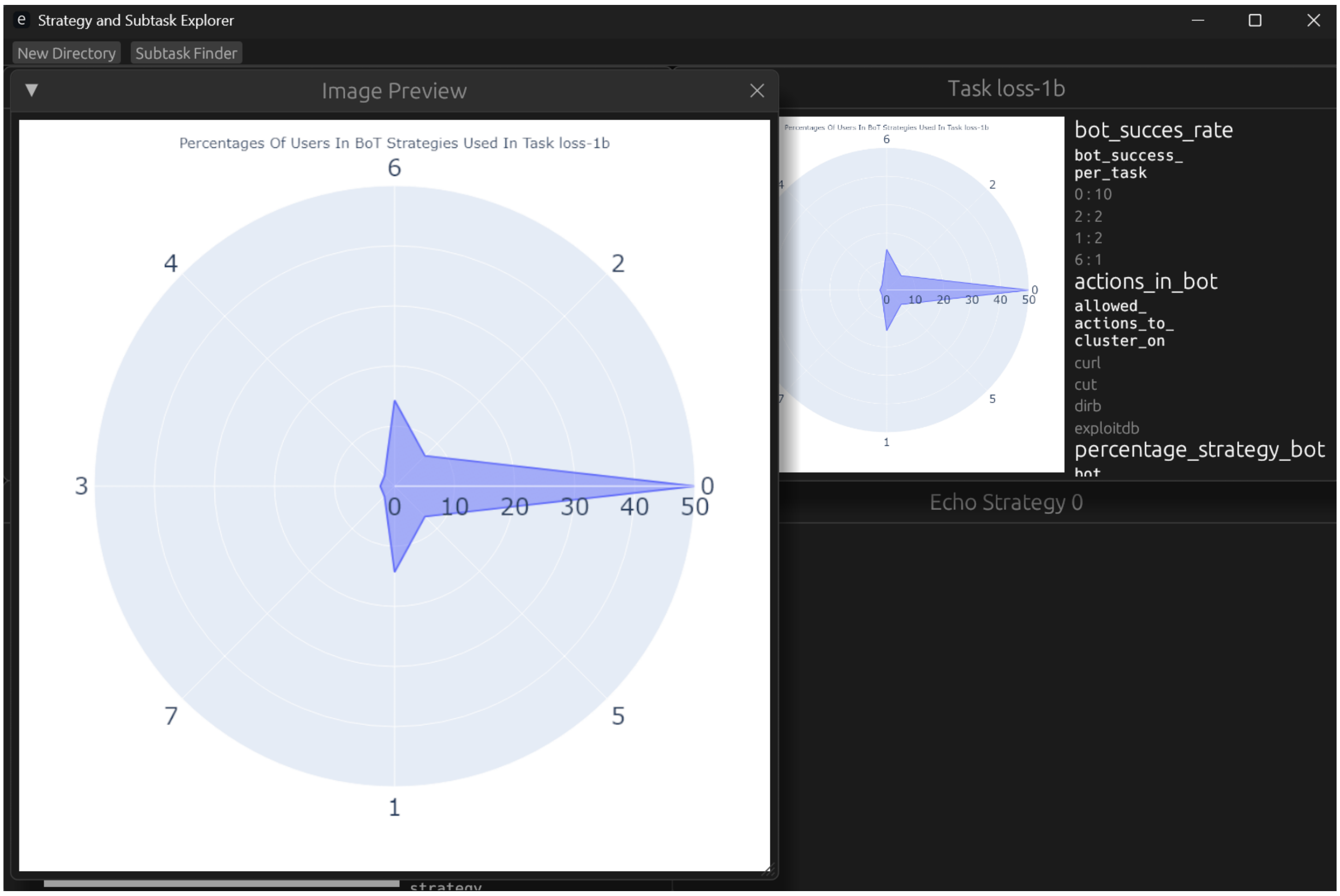
percentage_strategy_
echo

strategy

```
Figure 7: The Subtask Finder Pop−Up
```

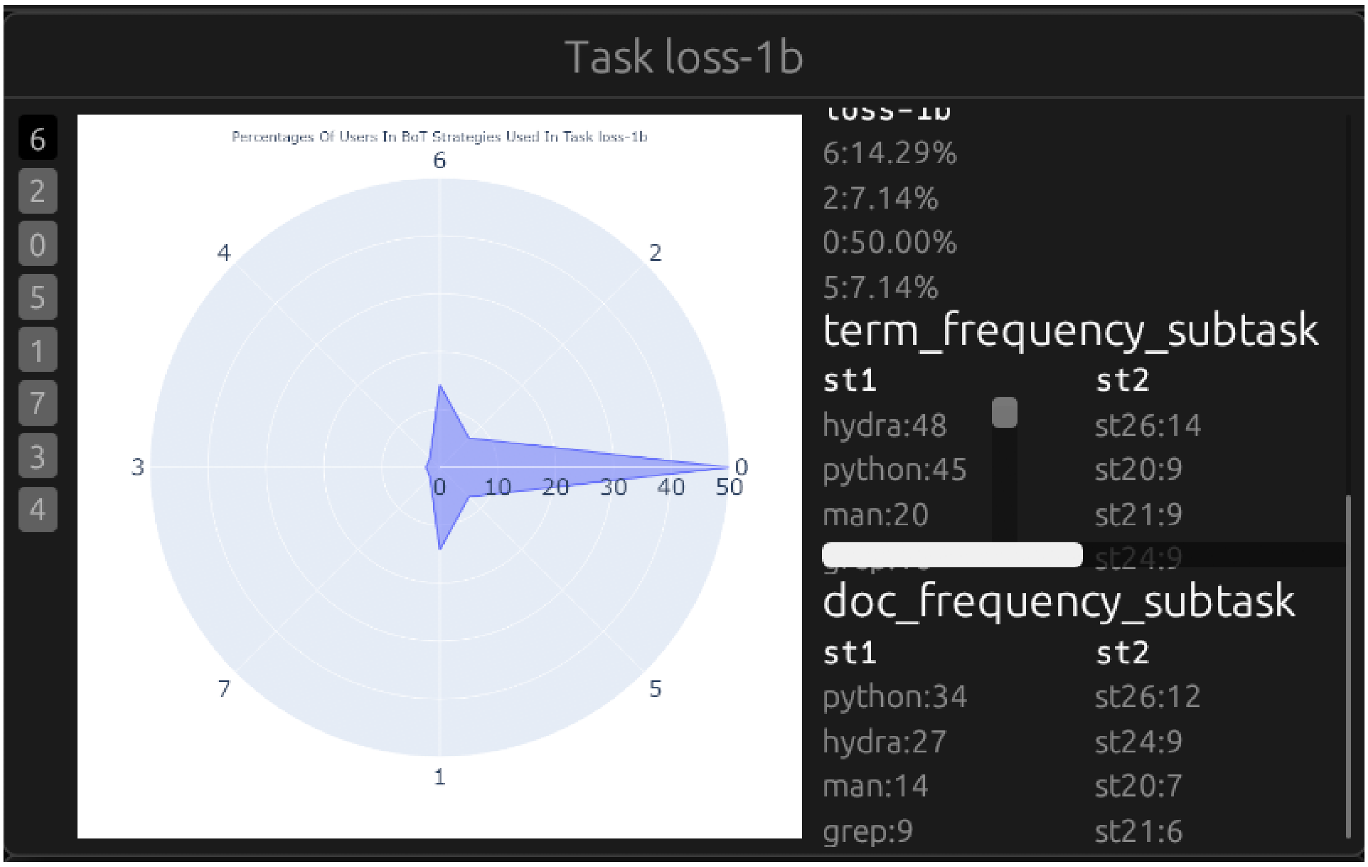
1.9 Other Features

If the user wants to focus on any displayed image, they can simply click on the image to summon the Image Preview pop-up (Figure 8). This displays the selected image in a pop-up that can be dragged and resized to view the image at different sizes. The pop-up also retains the image when selecting a different portion of the application, letting the user compare the image in the pop-up to other images that were updated by new selections.



```
Figure 8: The Image Preview Pop-Up
```

The statistics section in every panel is built to be infinitely scalable, allowing an infinite amount of statistics to be displayed. This was done to allow new statistics to be easily added at any time, without needing to update the application. Every statistic has a statistic type and a sub-statistic type. Only one instance of a statistic type can be displayed per panel, but infinite amounts of vertically scrolling sub-statistics can be displayed in a horizontal scrolling section within the statistic instance. Figure 9 shows an example of this scalable design, where ‘‘term_frequency_subtask’’ is the statistic instance of the ‘‘Action Term Frequency’’ statistic defined earlier. Inside of the statistic instance, there is a horizontal scrolling area for each sub-statistic type. The sub-statistics for term_frequency_subtask are the n-gram sizes for each subtask size. For each sub-statistic, there is a vertically scrolling instance listing each action/subtask and their term frequency.



loss-1b
6:14.29%
2:7.14%
0:50.00%
5:7.14%

term_frequency_subtask

st1	st2
hydra:48	st26:14
python:45	st20:9
man:20	st21:9
[illegible]	st24:9

doc_frequency_subtask

st1	st2
python:34	st26:12
hydra:27	st24:9
man:14	st20:7
grep:9	st21:6

```
Figure 9: An Example Of The Horizontal and Vertical Scrolling
Functionality In A Statistic Instance
```

To add more statistics, one must simply add the statistics for each statistic instance to thestatistics.jsonartifact file using the task explorer pipeline python script. An example statistic can be seen in Figure 10. From top to bottom, ‘‘task’’ tells the application panel to display the statistic in the Task panel, ‘‘term_frequency_subtask’’ is the unique statistic name, ‘‘st2’’ is the sub-statistic name, ‘‘loss-1b’’ is the task to display the statistic for, ‘‘st214:14|st20:9|st21:9 ...’’ is the statistics separated by the bar character, and ‘‘Subtask Term Frequency’’ is the user-friendly name of the statistic.

```
{

‘‘hierarchyLevel’’: ‘‘task’’,

‘‘statType’’: ‘‘term_frequency_subtask’’,

‘‘statSubtype’’: ‘‘st2’’,

‘‘identifier’’: ‘‘loss−1b’’,

‘‘statsList’’: ‘‘st214:14|st20:9|st21:9 ...’’,

‘‘header’’: ‘‘Subtask Term Frequency’’

}
```

```
Figure 10: An Example Of A Single Statistic Instance In Thestatistics.json
Artifact File
```

2 Interesting Statistics

With the information gained from identifying BoT strategies, echo strategies, and subtasks, there are many interesting statistics that can be computed to gain meaningful insight into how tasks are completed. There are many more interesting and helpful statistics that can be computed with the task explorer pipeline's results, but those listed are the statistics that were most prominently needed in our current research.

2.1 Action Run Frequency

Action run frequency is computed for every action and subtask in a task. It records the number of runs in which a given action or subtask was used. 39

2.2 Action Term Frequency

Action term frequency is computed for every action and subtask in a task. It records the total number of times a given action or subtask was used in the task.

2.3 Percentage Of Users In A Strategy

The percentage of users in a strategy is computed for every BoT and echo strategy in a task. This records the percentage of total users that are grouped within the strategy. The sum of all BoT or echo strategy percentages would equal 100%. The sum of all echo strategies in a BoT strategy would equal the percentage of users in the BoT strategy.

2.4 Descriptions & Side Effects

Descriptions and side effects are recorded for every run and subtask in a task. A description is a manually defined description of what an action does, written as if it was preceeded by the phrase ′′the user attempted to...′′. For example, the description for ′′hydra′′ is ′′use parallelized brute force password cracking on an online network protocol′′. When computed for a run or subtask, all descriptions for each action present are gathered into a single string description that describes what the user did. A side effect is a manually defined consequence of what an action does. For example, the side effect for ′′hydra′′ is ′′access gain to online network protocol′′. When computed for a run or subtask, all side effects for each action present are gathered into a string of side effects that lists the consequences of the run/subtask.

2.5 Success Rate

Success rate is computed for every BoT and echo strategy in a task. It records the number of runs in the strategy that lead to a successful completion of the task.

2.6 Actions In BoT

Actions in BoT is computed for every BoT strategy in a task. It lists the actions that were allowed when finding BoT strategies during factor analysis clustering.