MEDB STEM Conference 2016
Quack-a-thon

Declaration of Judging Criteria

Introduction:
Participants have a choice between competing in two separate but related areas or tracks for the Quack-a-thon. Regardless of the track chosen, each team will prepare a presentation on their work.

App Development Track:
The app development track focuses on the creation of applications with a security themed purpose using the Rubber Ducky hardware and software utilities. The goal is to produce useful tools from the viewpoint of security or otherwise respectable professions.[1] A demonstration has been prepared for students at http://uhmc.github.io/Ducky_Demo/

Policy Development Track:
The policy development track focuses on the creation of new policies given the scenario of being a school's administrator who come in contact with a Rubber Ducky around the school. The policy should enforce privacy, integrity and proper use of the facility in context of a Rubber Ducky.

The policies should cover the following situations:

1. Promoting ethical online behavior
2. Promoting Safe computing practices
3. Preventing social engineering tactics using stealth tools such as the USB Rubber Ducky
4. Controlling and preventing impact from malware and viruses, as well as reducing malicious activities such as DDOS attacks and others
5. Reducing or preventing online harassment, including sexting

Presentation:
Using a slide-show application of their choice, each team will be required to prepare a presentation containing the following elements:

- 1-2 slide title and introduction
- 1-3 slide team-member introduction and overview of tasks
- 3-5 slide policy OR app synopsis

[1] Malicious applications are in complete disagreement with the purpose of this competition and may incur disqualification depending on the severity and apparent malicious intent.

- 2-5 slide implications, implementation, and reasoning

Grading Rubric Overview:
Teams participating in the 2016 Quack-a-thon will be judged by the following criteria through the use of the accompanying rubric.

There are two dimensions contributing to a team's final score. The rows correspond to a list of attributes pertaining to a team's performance with a goal of allowing differently performing teams to have the same opportunity to earn points. The columns group different degrees of satisfaction with respect to the corresponding attributes. The farther right a check is placed for a corresponding attribute, the more points the team earns.

The far left column and bottom row are special in that they do not contribute points to a team's score in the way the others do. If an attribute earns a check in the leftmost column (excepting in the last row's case), it earns no points for the team. If any column other than the far right is checked for the last attribute (bottom row), a heavy penalty is applied to the team's score, increasing the farther left the check is placed and potentially disqualifying the team.

Results

App: Each team will decide on the outcome of the program. If the program successfully completes the goal, they will receive full credit.

Policy: The policy should be clear, concise and relevant.

Teamwork

App: Teams should divide the work equally among their peers, each member should be participating in order to perform well in this category.

Policy: Every member of the team should provide input or create research towards the policy chosen to receive full credit.

Documentation

App: The code should be properly documented in a text file within the Rubber Ducky for the judges to read. Code should also be commented.

Policy: The policy should be well presented and documented. There should be a clear explanation of why it should be put in place and what it would prevent to receive full credit.

Originality

App: Ideally, the program should have code written completely custom, however, teams can use pre-existing source code and do their own modifications or extensions.

Policy: A policy should be completely custom, however, points will still be awarded for the extensions and improvement of pre-existing policies.

Destructivity: Code should not be malicious or destructive, the team with an app that can cause damage will lose points depending on the severity.

Trait (% Weight)	Unsatisfactory (1 points)	Satisfactory (2 point)	Better (3 points)	Exceptional (4 points)
Results - 20%	No results.	The app produces some results but not as advertised. Policy is not clear or relevant.	The app produces most of the results advertised. Policy has a relevant goal, but has little real-world application.	The app produces all of the results advertised. Policy has a clear goal, is relevant, and has a good real-world application.
Teamwork - 20%	Almost no collaboration or one person did everything.	Most members participated.	Every member of the team participated.	Every member of the team participated, and tasks were divided between the members.
Documentation - 40%	No app documentation and/or the code is poorly organized and very difficult to read. Policy is badly explained, statements are empty with no evidence. Construction is skeletal, a rough-draft.	The documentation is simply comments embedded in the code. Policy is poorly explained, lacks details.	The documentation is embedded in the comments and contains a header that explains the app. Policy is explained well, but lacks substance such as evidence or an example to go with it.	There is another file documenting the app and is well written. Code should have a header and comments Policy is explained well. Statements are backed up by real world examples, or concrete evidence.

Originality - 20%	The app was not written by the team, a copy or almost a copy Policy was completely copied and pasted from somewhere; no real thought put into it.	The app employs a pre-existing concept and uses significantly modified code from an outside source. Policy was taken from a pre-existing policy but was lightly edited.	The app employs a pre-existing concept and uses no source code from outside sources. Policy was taken from a pre-existing policy and was extended with relevant points.	The app employs an original concept and uses no source code from outside sources. Policy was completely custom made, constructed and made solid through a valid framework of ideas.
	-3 points (disqualified)	**-2 points**	**-1 points**	**-0 points**
Destructivity (-100%)	Complete destruction of the system, persistent lockdowns, persistent forkbombs, anything that would decommission the judge's computer.	Temporary forkbombs, keyloggers, backdoors.	Nuisance, change monitor orientation, turn off wireless/ethernet something that would not destroy the system but would cause playful damage.	No malevolent code.