Launch Report - Cycle 1

For

FreeEDR

Submitted by

FreeEDR

Instructor:	Professor Greg Hislop
Team Members:	Bryan Bolesta, Ryan Fiers, Declan Kelly, Matthew Horger, Layla Phills, Zachary Santoro, Marisa Tranchitella
Cycle:	1
Date Submitted:	10/06/2019

Document template copyright 2005-2019, Gregory W. Hislop. Version 2.3. Use permitted under Creative Commons license CC-BY-SA. See http://creativecommons.org/licenses/by-sa/4.0/

Grading Rubric - Launch Report - Cycle 1

This rubric outlines the grading criteria for this document. Note that the criteria represent a plan for grading. Change is possible, especially given the dynamic nature of this course. Any change will be applied consistently for the entire class.

Achievement	Minimal	Exemplary	Pts	Score
Content	Section(s) missing, not useful, inconsistent, or wrong.	Provides all relevant information correctly and with appropriate detail		
Project			30	
Team			10	
Plan			30	
Grammar and Spelling	Many serious mistakes in grammar or spelling	Grammar, punctuation, and spelling all correct	10	
Expression **Tone**	Hard to follow or poor word choices Tone not appropriate for technical writing	Clear and concise. A pleasure to read Tone is consistently professional	10	
Organization **Layout**	Information difficult to locate Layout is inconsistent, visually distracting, or hinders use	All information is easy to find and important points stand out Layout is attractive, consistent, and helps guide the reader	10	
Late Submission				
Total			100	

Launch Report - Cycle 1

This report documents the initial launch of the project. It includes the project and team names, and an overview description of the project. It also includes a summary of issues related to the project and to the team, and basic elements of a plan for cycle 1.

Project

Project Name: FreeEDR

Abstract: Organizations that don't have a significant security budget can find it difficult to include workstations in their risk monitoring scope. Solutions such as forwarding logs from all workstations in an organization can be expensive because most SIEMs are priced based on log ingestion. Other tools that implement EDR, also known as *Endpoint Detection and Response*, are just as expensive to maintain and expand as organizations grow. This project aims to setup a series of scripts which will allow organizations who don't have the ability to purchase or implement an enterprise solution for EDR to monitor workstations across their organization. This solution is aimed at small to mid-tier organizations running Windows OS workstations who are looking for a maintainable and cost-efficient platform for EDR in order to secure their technological footprint.

Feature Highlights

- No external vendors needed to deploy solution for organization
- PowerShell scripts that can be easily developed and managed by an IT Infrastructure team
- Cost-effective solution for organizations with a limited budget
- Solution is flexible as organization and number of workstations grows
- Solution requires limited knowledgebase for new team members in an organization to learn

External Stakeholder or Proxy User

Nick Ascoli

Security Risk Advisors, Senior Threat Management Consultant

nick.ascoli@securityriskadvisors.com

https://www.linkedin.com/in/nicholas-ascoli-28a78b93

Issues

No expected issues in regards to accessing information as this is an Open Source project. All work will be done via a private GitHub repository, which will either be made public or transferred to SRA at the completion of this project.

Team

Team Name: FreeEDR

Team ID: FreeEDR

Team Members and Roles

Name	Role	Initials
Bryan Bolesta	Developer	bb
Ryan Fiers	Developer	rf
Declan Kelly	Developer	dk
Matthew Horger	Source Control Coordinator, Documentation	mh
Layla Phills	QA/UAT Tester, Client Facing Support	lp
Zachary Santoro	Lead Developer	zs
Marisa Tranchitella	Project Manager	mt

Team Communication

Weekly Meeting: 5:30pm every Monday @ 3675 Market St

Group (Private) Repository: https://github.com/mhorger3/FreeEDR

Our team established a Discord server, managed by Marisa Tranchitella, our project manager. The discord server has built in voice communication options, but we can also collaborate via standard phone calls.

Some of our team members will communicate face-to-face if they choose to work on this project during their part-time work, or if we meet outside of our scheduled class times. Communication with our external stakeholder will be either face-to-face or via email.

Team Issues

Many of our team members work part-time jobs in addition to being full-time students. This will cause some issues in time management of this project when working towards deliverables especially when things get busy either at work or school. To alleviate this issue, 4 of our team members work at Security Risk Advisors so they have constant contact with our external stakeholder in case we have time issues getting in touch with him. These team members, along with two others who work in Risk Assessment and Infrastructure positions, will also be doing relevant work related to the project during their part-time jobs so they will not have forgotten any material.

Activity Plan

Objectives

The team's objectives for this cycle are:

- Establish Github Analysis tools, require RFC's (request for changes) to be made for merges to master
- Complete drafts of all project documentation and cycle deliverables with enough time for UAT acceptance and final product integration.
- Maintain clear communication with external stakeholder / CCI instructor if any team issues arise
- Start developing project implementation, which includes virtual machine creation for test deployments in the cloud.

Activities

The table below shows the initially identified set of activities for this cycle and assigns a lead person to each activity.

Week	Person	Contribution
2	mh	Initial Launch Cycle complete, Github setup
3	zs	Requirement specifications started, Github setup complete
4	lp	Test Specifications started, start architecting infrastructure/design
5	mt	Midway check-ins, resolve any outstanding deliverables or issues preventing us from further submissions
6	dk	Design document started, spin up virtual machines for development
7	lp	Test Specifications finished, start to compile client-facing documentation
8	dk	Design document finished, begin script development
9	zs	Submit client-facing documentation to external stakeholder for review
10	bb/rf	Continue implementation and development