FEASIBILITY REPORT

Gaston Carvallo - 048530133, Loyd Rafols - 022827158

REA605: Research Methodologies

March 28, 2021

Contents

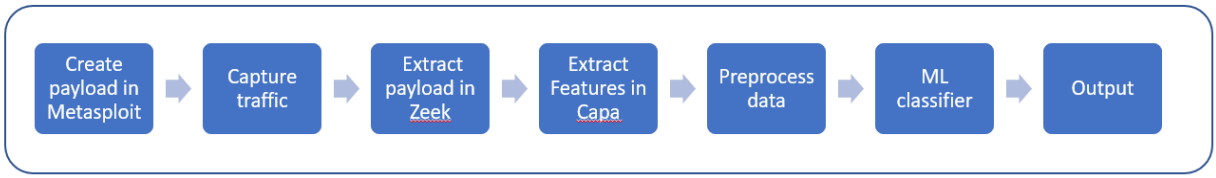
Introduction

This report serves to document our self-assessment regarding the feasibility of our research project. In summary, our research project is to create a practical introduction to detecting transmission of malicious payloads using machine learning. Our current plan is to generate payloads in Metasploit, have the "victim" download the payload, capture the connection with Zeek, then extract the features using Capa, then finally use scikit-learn to classify the data.

In this document, we will be looking at three aspects of feasibility for the project - technical (technologies and processes used), legal (software licenses), and schedule (time constraints, timeline of tasks).

Technical Feasibility

In general, when making decisions about the project implementation, priority was given on the use of technology that at least one of the participants had some familiarity with it. Our proposed flow looks like:



To accomplish this, we have identified the following high level technical tasks:

- Deploy test environment.
- Automation of the process to create the sample data.
- Create the scripts to extract features and preprocess the data
- Create the scripts to train and validate model
- Implementation of the classifier
- Output results to a log

The payloads will be created using the msfvenom module of Metasploit and we will extract the file from the Zeek file logs.

The only two technologies that we have not used before are Capa and the scikit-learn python library. One is a command line application and the other is a python library. So, we don't expect any major problem implementing either of those.

In our assessment, the project can be accomplished using python, and while not trivial, we consider it well within our expertise's and don't foresee being it an impediment to the successful completion of our project.

Legal Feasibility

In terms of legal feasibility, the main points of contention that may cause problems revolves around licenses by the software or tools we use.

Scikit-learn

According to documentation provided by PyPI (https://pypi.org/project/scikit-learn/), scikit-learn - a Python module we will use for leveraging machine learning in the project - uses a 3-clause BSD license which states the following stipulations for reproduction (https://opensource.org/licenses/BSD-3-Clause):

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

In summary, if the above clauses are following when building and configuring the learning environment, there will be no legal issues in misusing the license provided by scikit-learn. We believe this will have minimal impact on the feasibility or success of the project overall.

Metasploit Framework

The Metasploit Framework will be used as our primary means of generating malicious payloads to transmit across the network to our experiment test bed through the msfvenom tool. Like scikit-learn, the Metasploit Framework also uses a 3-clause BSD license with individual licenses as required for other third-party components (as outlined here: https://github.com/rapid7/metasploit-framework/blob/master/COPYING). Following the license stipulations for the Metasploit Framework is handled in the same manner as scikit-learn, so we do not believe this will compromise the feasibility or success of the project.

Capa

Capa is the tool we are using to extract the features in order to determine if a sample is malicious or benign. Capa is licensed using Apache 2.0, with the following stipulations according to section 4, “Redistribution” (https://github.com/fireeye/capa/blob/master/LICENSE.txt):

a. You must give any other recipients of the Work or Derivative Works a copy of this License; and
b. You must cause any modified files to carry prominent notices stating that You changed the files; and
c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

Therefore, the only requirement to legally use Capa is to include the Apache 2.0 license, which will need to be bundled in our training environment. However, this will not prove to be significant in affecting the feasibility and success of the project.

Zeek

Zeek will be used to analyze the network as samples are sent to the experiment testbed VM. Zeek utilizes the same 3-clause BSD license as scikit-learn, with a slight modification to change the copyright holder to those appropriate, such as UC Berkeley (https://github.com/zeek/zeek/blob/master/COPYING):

- Neither the name of the University of California, Lawrence Berkeley Laboratory, U.S. Dept. of Energy, International Computer Science Institute, nor the names of contributors may be used to endorse or promote products derived from this software without specific prior permission.

We believe these licensing restrictions will not have a significant impact on the feasibility or success of the project.

Windows

Windows is our selected operating system for running our curriculum, generating data, and running the machine learning algorithms. Specifically, we plan on using some version of Windows 10. Windows does not use an open license; specifically, Microsoft has specified the following for licensing (https://www.microsoft.com/en-us/Useterms/Retail/Windows/10/UseTerms_Retail_Windows_10_English.htm):

- Section 2.a: The software is licensed, not sold. Under this agreement, we grant you the right to install and run one instance of the software on your device (the licensed device), for use by one person at a time, so long as you comply with all the terms of this agreement. Updating or upgrading from non-genuine software with software from Microsoft or authorized sources does not make your original version or the updated/upgraded version genuine, and in that situation, you do not have a license to use the software.
- Section 2.c: The device manufacturer or installer and Microsoft reserve all rights (such as rights under intellectual property laws) not expressly granted in this agreement. For example, this license does not give you any right to, and you may not:
    - (i) use or virtualize features of the software separately;
    - (ii) publish, copy (other than the permitted backup copy), rent, lease, or lend the software;
    - (iii) transfer the software (except as permitted by this agreement);
    - (iv) work around any technical restrictions or limitations in the software;
    - (v) use the software as server software, for commercial hosting, make the software available for simultaneous use by multiple users over a network, install the software on a server and allow users to access it remotely, or install the software on a device for use only by remote users;
    - (vi) reverse engineer, decompile, or disassemble the software, or attempt to do so, except and only to the extent that the foregoing restriction is (a) permitted by applicable law; (b) permitted by licensing terms governing the use of open-source components that may be included with the software; or (c) required to debug changes to any libraries licensed under the GNU Lesser General Public License which are included with and linked to by the software; and
    - (vii) when using Internet-based features you may not use those features in any way that could interfere with anyone else's use of them, or to try to gain access to or use any service, data, account, or network, in an unauthorized manner.

In summary of this section of the Windows 10 license, as long as we do not distribute a *licensed* copy of Windows 10, we are permitted to distribute an *unlicensed* copy of Windows 10 as an OVA file. We do not believe that any features provided by a licensed copy of Windows 10 is required for any of the software or tasks done during the project or required for the final deliverable, so this license will have little impact on the feasibility or success of the project.

Other areas of concern

Regarding experimentation, since we are performing experiments in a self-contained environment with no need for human subjects - aside from perhaps feedback in the later stages of the project independent of the experiments, the regulations outlined in TCPS 2 are not a large concern for the legal feasibility of the project.

Summary

In summary, the legal feasibility of the project revolves around licensing for the software we plan to use. These software licenses include, but is not limited to:

- Scikit-learn (3-clause BSD)
- Metasploit Framework (3-clause BSD)
- Capa (Apache 2.0)
- Zeek (BSD)
- Windows 10 (Custom)

We believe these licenses will not prove to be an obstacle when implementing the project and do not affect the short- or long-term feasibility of the project. Other legal areas of concern we identified were in the experimentation phase, but since we are not utilizing human subjects as part of the experiment, we believe that the feasibility of the project is not compromised by this.

Schedule Feasibility

In scheduling our project, we have assumed 12 weeks of work in each semester, of which there are two - semester 1 being REA705, and semester 2 being REA820. A high-level view of our project plan is to complete the technical implementation during semester 1 (REA705), and to write the final report, labs and complete the final deliverable before the end of semester 2 (REA820).

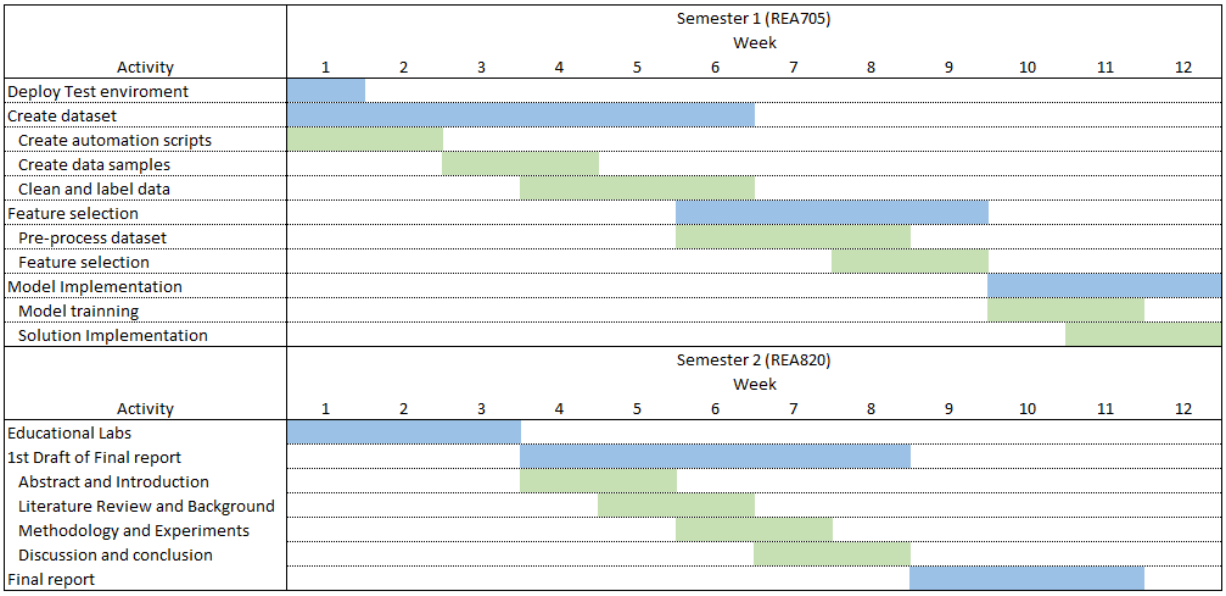
While dependencies tend to be finish-to-start, in some cases a lead can be applied, and the next task can start before the previous one has finished. For example, while we cannot pre-process the sample data until its collected, certain preliminary activities around it can be started once we have information about the output format for the dataset has been established. Another example is configuring most aspects of the learning environment (ie. Windows desktop) in advance before the technical implementation is completed.

We will create a detailed project management plan with our final proposal, including all low-level tasks, each assigned to a specific person and hold at least weekly team meetings during the

project implementation to make sure execution is being completed as planned and discuss potential problems on upcoming tasks.

We are confident that the initial project plan doesn't present significant risk on completion, and that if needed additional parallelization can be applied to finish the project on time.

HIGH LEVEL PROJECT PLAN



Conclusion

In conclusion, we reviewed the technical, legal and schedule aspects of feasibility for our project. For technical feasibility, we explained high-level tasks that will help us achieve our final goal, and which tools we would use to reach our final goal. For legal feasibility, we reviewed the licensing clauses required by the software used as part of the project as a whole, and those that will be shared as part of the educational environment. Finally, for the schedule feasibility, we explained that given the complexity of this project, we do not expect issues with the project timeline and having the final deliverable be completed on time.