NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

Midterm Assignment

Course: *Information Security and Privacy*
Professor: *Kevin Gallagher*
Due date: *August 21st, 2020*

1. Goals

In this midterm we will be looking at solidifying the concept of the security mindset. We will do this using a case-study of an open source software project wishing to protect itself from malicious actors. By the end of this assignment you will have:

1. Identified the assets held by an example (though fictitious) open source software project. See section 3.
2. Created an attack tree to identify the easiest and most effective attack. See section 4.
3. Stepped through the cyber kill-chain to describe the details of how an attack against the project could occur. See section 5.
4. Applied knowledge of the cryptographic primitives (hashes, signatures, encryption, PKI) to defend against the attack that you considered in step 3. See section 6.
5. Described how TUF helps update security using the cyber-killchain. See section 7

Each of these goals is described in further sections.

2. Background

In this assignment we will be considering a case-study involving an open source software project. This software project creates software that is meant to be pushed out to end users. In this specific example it doesn't really matter what the software does, other than to know that the software does not require root on the user's computer.

The project stores all of its code on GitHub. Because the project is open source, all code is available for inspection by anyone. The project has a small set of 5 contributors who have access to the GitHub account and can push straight to the repository, review pull requests, create new branches, etc.

After the code is ready for release the project builds binaries and puts them on their own websites. This is to distribute the software to new users.

For users who are already using their software, the project also has an update server that the software connects to when it begins to run. It checks if a new version is available, and if so, downloads and installs the update.

3. Identifying assets

In order to move forward with threat modeling we first need an understanding of what assets the project has that an attacker may be interested in. Keeping these assets in mind allows us to consider what attacks may be more likely and therefore create more accurate attack trees.

For this part of the assignment, please list the assets that the project owns, controls, could control, or has access to and describe why an adversary would be interested in getting control of that asset. Then rank all of the assets you listed in order of most desirable to an attacker to least desirable to an attacker.

4. Attack Trees

Now that we have identified the assets that attacks may be after, we would like to think about potential attacks to get to those assets. For the purpose of this assignment, we want to think about the computers that the project's software runs on as the asset that the attacker is most interested in. Said another way, the attacker wants to be able to run code on the computers owned by the software project's users.

With this in mind, create an attack tree to demonstrate how an attacker can achieve this goal. When the attack tree is finished, label each leaf node as easy, medium, or difficult. Then propagate this information up the tree.

5. Kill Chain

One of the potential attack vectors to run arbitrary code on users' computers is through the update server. If the attacker can poison the code update, they can distribute their malicious code to all current users of the software. In this section you will be explaining all steps of this attack using the framework of the cyber killchain.

For each step in the cyber killchain, describe what is available to the attacker, what the attacker needs to achieve, and how the attacker can be stopped in performing their intended action.

6. Applying Cryptography

Hopefully the previous exercises have demonstrated that there are many avenues open for an attacker who wants to gain access to assets controlled by an organization. Now that we know what some of these avenues are, we want to begin applying some of the tools that we learned about to make it more difficult for attackers to achieve their goal. One of the strongest tools in the defender toolbox are cryptographic primitives, such as encryption, hashing, MACs, digital signatures, and PKI.

Using the primitives mentioned above, please describe how you would design the update mechanism to thwart an attacker looking to use the update system to run arbitrary code on users' machines. You may use diagrams, tables, or any other visual aid.

7. Comparing to TUF

In Part 4 of this assignment you developed a solution to the issues of update security for an open source project using some cryptographic primitives. In this section you will read about The Update Framework (TUF) developed by Professor Capos' Secure Systems Lab.

Please compare and contrast your approach with the approach taken by TUF. What do you do similarly? What do you do differently, and why? Which attacks did TUF consider that you did not?

8. Closing Remarks

All projects should be turned in via NYU Classes.