**Slide 1**:

Good morning, today I am going to talk about the research titled Explanation-Guided Backdoor Poisoning Attacks against Malware Classifiers.

**Slide 2:**

Lets go over some prerequisite information.

In machine learning models, especially in classifiers, the model learns about classification from different features. For example, if we look at a face detection classifier, the features it uses will be like gradient, edges, color ways and so on. The classifier focuses on these features when given an input to make a decision.

Malware classifiers, work similarly where currently the malware classifiers utilize static analysis to detect malware in an executable. Static analysis is a method of using program binaries to understand the functioning of the code and determine some information about the program. This helps malware classifiers not be exposed to attacks as the program is never executed.

Explanation guided machine learning. In early machine learning days, it was difficult to understand what features the model learns from if the model was unsupervised, which makes the trust of the model in areas like security lesser. The use of explanation analysis models like SHAP help us identify features in model which contribute to the decision-making of the model.

Backdoor attacks, lets watch a video.  
  
**Slide 4:**

As machine learning is becoming integral in all aspects, it has become important in security. Malware classifiers using machine learning techniques to classify different executables into either malware or goodware.

The research sees this as an opportunity to introduce poisoning attacks on these classifiers. The research aims to explore poisoning of classifiers in such a way that they can install a backdoor in the classifier to evade detection of some type of classifier.

The research explores the effectiveness of these attacks in malware classification.

Furthermore, the research uses SHAP which is an explanation model to select features that would help in creation of a watermark in a model-agnostic way.

**Slide 5:**

ML-Based Malware classifer are developed in different steps. Firstly, the data is acquired from users. This is done so that there is no overfitting issue when training the model. The user data is the combined with the data that is proprietary and previously tested. The datasets are then merged and randomized. The classifier model is then trained over this dataset. Training involves feature extraction, Machine Learning training, and classification methods.

These could be as simple as using Euclidean distance to classify different things, given one classification or complex with the use of deep convolution layers.

**Slide 6:**

The threat model of the research is to be able to create a new classification in malware classifiers such that it differs from a clean classifer, however, evades detection techniques that would classify it as malware.

The developed model should lead to predictions that were chosen by the attackers, however, maintain normal response to clean inputs.

This attack threat model will then be evaluated on different levels of restrictions.

**Slide 7:**

First, there should be an established control over the feature subsets to induce a new classification. Given that the malware classifiers use data from users, however, create different classifications on themselves, the adversary wants to control features in a way that they can make the decision for their input always go through the same classification class using poisoned values.

This involves creating an area in the feature that would influence the decision boundary of the classifier. Making an area away from the center, however, still being close enough to make sure it is always classified as goodware.

The features are selected after running SHAP over the classifier models. The SHAP output helps identify features that most heavily affect the decision of a given input.

The attack would use independent selection techniques for maximizing attack effect. This means that feature in this phase would be selected for making sure the attack has maximum impact.

The second technique is to use Greedy-combined selection for creation of watermark with goodware features.

So, essentially after selecting the features, the inputs are created in such a way that have malicious functionality, however, they include a distinct watermark that would manipulate the decision making of the classifer.

**Slide 8:**

The first evaluation was done in a controlled environment where the dataset used was EMBER which has 1.1 million Portable Executables.

The classifier models it was evaluated over were EmberNN and LightGBM.

When the LightGBM model was attacked, the results were effective with creating a misclassification with small trigger size. (Trigger size is essentially the pattern that causes misclassification, same as watermark).

EmberNN was more resilient as it used deeper connected layers which produce more effective results and can prune the outliers, which is where the watermark was developed. However, EmberNN could be attacked if the trigger size was increased. (Meaning the complexity of the water mark would have to increase to be able to manipulate the classifier.

**Slide 9:**

The attack was then conducted on windows by using a library called pefile. The pefile was used to create a backdoor trigger to windows binaries.

As windows uses EmberNN, the attack had to be developed with selective features. SHAP was utilized to identify the features that were not related to hashing and directly editable. These features were good targets for basing the water mark. Before development of the watermark, the features that had interdependency were pruned to remove computational overhead.

**Slide 10:**

The attack that was developed showed significant effectiveness even when the features that would manipulate the decision were as low as 17. ‘

Even when a black box approach, which means a blind approach, where we do not have information about classifiers and their functioning, was effective with high accuracy. This approach is computationally heavy, but considered feasible as feature detection is not required constantly, but only once.

Lastly, even when the program executables are run, the watermarks retained its label and its malicious functionality, stating that not only can the malware be loaded but executed also.

**Slide 11:**

The paper was significant as it helped identify a new vulnerability in the field of cyber security where backdoors can be installed for clean classification leading attackers to be able to install malware on systems without defense mechanisms taking place.

It proposed a generalized solution for installation of backdoors in feature based classifiers.

Lastly, even under realistic evaluation constraints, the model proves practical feasibility and effectiveness.

**Slide 12:**

The paper concluded with some mitigation techniques,

Firstly, the use of spectral signatures, where samples with high outlier score of one singular value should be pruned. So, if the watermark is based on manipulation of one feature, the samples would be pruned and classified as malware.

Secondly, Isolation forest which uses the same outlier concept however, the detection instead of being on one single feature, depends on anomaly detection.

The only limitation of this attack so far is the attacker’s knowledge on feature space of victim model which can be by passed as shown by the black box testing.