Application of Data Mining Classification Techniques to Supply Chain Attacks Dataset

Rimma Shilkina   
 Computer Science Department of College of Engineering  
 University of North Texas Denton, TX, USA

rimmashilkina@my.unt.edu

***Abstract***

Cybersecurity and, particularly, Supply Chain Attacks, are an important and widely discussed topics these days, but especially after SolarWinds attack in 2020. There is a significant lack of research papers related to applying Data Mining techniques to Supply Chain Attacks data, mostly because of attacks’ information unavailability for public. The goal of the project is to apply a classification technique to a Supply Chain Attack dataset by building models in Java and Weka, to classify these attacks based on the Attacker Type, to compare their performance, and to use these models to predict future attacks like SolarWinds 2020. For this purpose, Decision Tree algorithm was used both for Java code and Weka. Comparison of the models revealed better classification performance of Java model and the same prediction result was made for the SolarWinds attack.

***Keywords***

Cybersecurity, Supply Chain Attack, Classification, Data Mining

***Introduction***

According to the definition of the National Institute of Standards and Technology, Supply Chain Attacks (also called Third-Party Attacks) are “attacks that allow the adversary to utilize implants or other vulnerabilities inserted prior to installation in order to infiltrate data, or manipulate information technology hardware, software, operating systems, peripherals (information technology products) or services at any point during the life”1.

Risks related to supply chain attacks significantly increased during last few years as more enterprises’ data became available to suppliers and service providers: 78% increase in 2018 compared to 20172, almost doubled in 2019, and 430% increase in 20203. Most known attacks happened within last few years include: Target, a US retailer, (2013) where hackers breached Point of Sales (POS) systems of 1800 stores and stole credit and debit cards data of 40 million customers, the overall cost of recovering was $61 million; NotPetya (2017) is a malicious code inserted into updates of the financial package "M.E.Doc" - tax preparation software for the Ukraine - estimated more than $10 billion in total damages worldwide4; SolarWinds (2020) about 18,000 customers were affected, including Commerce, Treasury, Homeland Security and Justice Departments of the US government as well as Microsoft, FireEye, and other private companies.

The purpose of the project is to build a classification model that can identify adversaries as “State”, “Criminal”, “N/A”, or “Unknown” based on subset of attributes (most relevant attributes was chosen based on own knowledge and two research papers (discussed below)), compare it with dataset classification (the attribute “Attacker Type” was chosen as a class label); add the Solar Winds attack to the dataset and predict its label.

***Background and Literature Review***

The most relevant to current topic and publicly available paper is the “Supply Chain Attack Framework and Attack Patterns” by John F. Miller that was published in December 20135. As stated in the paper, it is the first framework which combined data from all available resources (at the time of the paper writing) to assist the Department of Defense to “understand the nature and potential extent of supply chain attacks”. Based on 12 attributes describing supply chain attacks, the author created 41 attack patterns which can be used “to support threat analysis and vulnerability assessments”. There is no description of what techniques and methods were used to discover those patterns. The additional part of the work was to create a catalog of initial countermeasures that can be used after an attack pattern was identified. This catalog includes 20 countermeasure’s descriptions and recommendations.

In 2019 a group of UK cybersecurity researchers published their work “Managing cyber risk in supply chains: A review and research agenda” 6. The purpose of the research was to answer to a question “How can organizations manage cyber risks in supply chains?”. They used a Systematic Literature Review as the main approach for screening and assessing quality of academic articles published from 1997 to 2017. Text mining and connectivity-based clustering were used to create typology for data analysis - five core themes (cyber risk types, cyber risk propagation, cyber risk points of penetration, cyber security challenges and mitigation measures). The study proposes a Supply Chain Cyber Security System - a conceptual model that integrates IT, organization and supply chain security systems and emphasizes how it is essential for “successful implementation of cyber risk mitigation strategies”.

Both articles were useful for the Project from two points: first, analyzing the attributes for the better attacks’ classification; second, looking at the used data mining techniques (the second paper).

***Dataset characteristics***

The main challenge for any data mining research in cybersecurity area is lack of publicly available data because of unwillingness to disclose it by companies. In 2020 the Atlantic Council, an American think tank in the field of international affairs, presented a project “Breaking Trust” - a dataset of 115 incidents - 82 attacks and 33 disclosures - from 2010-2020 years with 17 attributes:

* Date: Best estimated start date of the attack or discovery date.
* Name: name of the attack.
* Attack/Disclosure: Whether the entry is a verified attack or disclosed vulnerability.
* Summary: of the entry, its impact, and technical details.
* Article(s): Links to article(s) about the entry.
* Affected Code: What code was modified by attackers, or what code had a vulnerability in it.
* Code Location/Owner: Who owned that code, or, if open source, the repository name.
* Downstream Target: The end-target of the attack.
* Affected Codebase: Categories describing the codebase, product, or service modified by attackers (1st Party OS/ Applications, 1st Party Firmware, 3rd Party Application, 3rd Party Firmware, OSS (Open-source software), Attacker Application, or Unknown/NA).
* Attack Vector: How the attacker was able to edit the affected code without detection (Stolen/purchased certificate, Pre-signature insertion, i.e., Attacker modified code before it was signed by developer, Default password exploit, Account access, Self-signed/Unsigned, i.e., Attacker signed software, or no signature required, Broken signature system, Unknown, Other, or N/A).
* Distribution Vector: How the attacker was able to distribute the modified code (Typosquatting, Hijacked updates, Proprietary application store, 3rd party application store, Open-source dependency, Worm component, Hardware component, Direct download, Phishing, Development software, Supply chain service provider, Unknown, other, or N/A)
* Supply Chain Potential (Credential theft, Certificate theft, Cryptography Error, Firmware Editing, Default password, Code injection, N/A).
* Impact (Data extraction, Physical systems, Backdoor access, Cryptominer, Remote command execution/ download, Adware, Payment diversion, Establish botnet, Data damage, Unknown).
* Attacker Name.
* Attacker Type (State, Criminal, Other, Unknown, or N/A).
* Infectious Potential: The degree of proximity a given piece of malware needs to spread to a target, from requiring physical access to a piece of hardware, all the way down to worm capabilities allowing it to proliferate through networks autonomously: 0 - Physical access, 1 - Product components, 2 - Store download, 3 - Network connection, 4 - Worm.
* Depth of Impact: The level of meaningful and harmful access a given piece of malware has in an infected machine, from an infected application all the way down to the capability to damage physical systems: 1 – Application, 2 – Update, 3 – OS, 4 – Firmware, 5 - Physical system.

The goal of the Atlantic Council was to consolidate and visualize data related to supply chain attacks.

***Data preprocessing and Transformation***

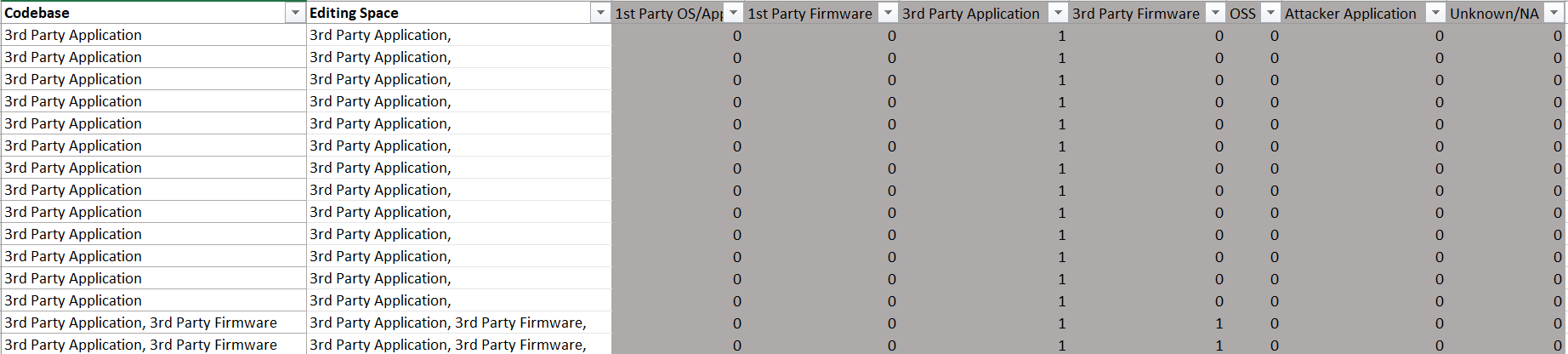
3 stages of the data preprocessing were the attribute selection, converting values of attributes into a numeric format, and removal of “Other” category from the attribute which is used as a class label.

In the second part of the Project, the SolarWinds 2020 attack was added to the original dataset to check predictive abilities of models.

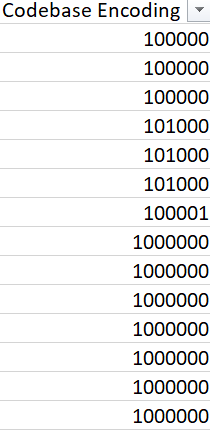
Attribute Selection was based on a common sense, knowledge, and research papers which were introduced earlier. 5 attributes were chosen from 17:

* Codebase
* Attack vector
* Distribution vector
* Supply Chain potential
* Impact

All chosen attributes are multivalued in the original dataset, they have additional sub-attributes that are used to generate them. Please, see how it looks on a screenshot below:



To work with the dataset, the “Editing Space” column was eliminated, and sub-attributes were joined into one attribute with numeric representation of the data (an Excel formula was used). The Codebase attribute after manipulation with data:



The Attack vector, Distribution vector, Supply Chain Potential, and Impact attributes were converted to the same format.

The class label attribute is the Attacker Type. The importance of this attribute is to predict whether the attacker is State or Criminal. Tools, tactics, and procedures (TTPs) could be very similar between these actors. But Criminals usually seek some immediate monetary gain, while state sponsored adversaries pursue long time goals: espionage, sabotage, etc. State actors would drop a lot of tools to maintain their access: if one tool is discovered, there might be few other tools just waiting commands from their operators. In other words, “depth” of the attack and consecutive damages are more severe from the state attacks than form criminal (although, it depends on a particular case).

The Attacker Type attribute has five possible values: State, Criminal, N/A (for disclosures), Unknown (no public attribution in cited sources) and Other (any group not clearly state or criminal - e.g., ideological nonstate, researcher, etc.). After first attempt to build a classification model by using the Attacker Type attribute, the model was not able to correctly classify the “Other” label. There were only three records with the “Other” Attacker Type among 115 records. To get better model and because of insignificant number of records with the “Other” label (2.6% from total), the records with the “Other” label were deleted from the dataset.

To predict class label for the SolarWinds 2020 attack, a separate CSV-file with one line which describes it was created. Parameters were chosen based on available information.

Attackers used SolarWinds Orion network management software and inserted their own code into it. SolarWinds was not a target, but its customers were. From affected customers point of view, SolarWinds Orion is 3rd party application (software):

|  |  |
| --- | --- |
| **Editing Space Codebase** | **Codebase Encoding** |
| 3rd Party Application, Attacker Application, | 10010 |

Attackers were able to insert their code during build process of Orion’s updates. This allowed them to insert malware just before the Orion build platform digitally signs (pre-signature insertion) new updates. Attackers were able to obtain account information for build subsystem and to break into digital signature system.

|  |  |
| --- | --- |
| **Editing Space Attack Vector** | **Attack Vector Encoding** |
| Pre-signature Insertion; Account Access; Broken Signature System; | 101010 |

After they successfully inserted malicious software (malware), this malware was distributed via normal (regular) update process to SolarWinds Customers. SolarWinds here is Supply Chain Service Provider. SolarWinds build platform was broken and it distributed malicious updates.

|  |  |
| --- | --- |
| **Editing Space Distribution Vector** | **DV Encoding** |
| Hijacked Updates; Development Software; Supply Chain Service Provider; | 10000000110 |

After malware was initiated at customers’ networks, attackers were able to perform various tasks including theft of credentials. They also placed a lot of other tools to maintain their access to penetrated networks in case some tools would be discovered.

|  |  |
| --- | --- |
| **Editing Space Supply Chain Potential** | **SCP Encoding** |
| Credential Theft; Code Injection; | 1000010 |

Impact of this attack is not known to full extent. They were able to roam affected networks for months, gathering information and sending it to their servers. They left a lot of backdoors (there is a possibility that some are still undiscovered) to keep their access.

|  |  |
| --- | --- |
| **Editing Space Impact** | **Impact Encoding** |
| Data Extraction; Backdoor Access; | 1010000000 |

Intelligence and security community believes that the attacker is a Russian state-sponsored group known as APT29 or Cozy Bear. It is supposed to be a part of SVR (Foreign Intelligence Service)

|  |  |
| --- | --- |
| **Attacker Name** | **Attacker Type** |
| APT29 / Cozy Bear / SVR | State |

***Data Mining - Building Classification Model***

“Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels.”7

Every classification model is a two-step process. First, it is a training step - constructing the model by learning the data and finding ways to classify it. Second, it is a label prediction. The goal of the project is to classify supply chain attacks described in the dataset as “State”, “Criminal”, “Unknown” or “N/A”.

There are several techniques that can be used for the classification such as Decision Tree based methods, K-nearest Neighbor Classifier,

Neural Networks, Naive Bayes and Bayesian Belief Networks and others. The Decision Tree classifier was chosen for the project.

Two independent classification models were built for a given dataset. The first one is a model based on Java algorithm. Weka was used for building the second model. Weka is a collection of machine learning algorithms for data mining tasks.

*Decision Tree Classification Model created with Java application.*

The program works as follows:

1. Loads a dataset from a CSV-file (cs\_data.csv).

2. Builds (trains) a decision tree. 66% of data points were used to train the model.

3. Tests a tree on the loaded dataset (test was performed on another 34% of data points).

4. Loads another file with new parameters (for the SolarWinds 2020 attack).

5. Runs these parameters against the created decision tree.

The decision tree build algorithm:

1. Calculates initial entropy for all class labels.

2. Selects one independent variable, maps all strings that contain this variable into a map structure.

3. Calculates entropy for this variable and calculates an information gain.

4. If new information gain is greater than previous, it splits dataset into two parts, and records a splitting point.

5. It evaluates left part and right part independently (recursively) after splitting.

6. If the algorithm cannot calculate new split point on step 4, it returns to the step 2 (selects another variable and starts decision tree build algorithm).

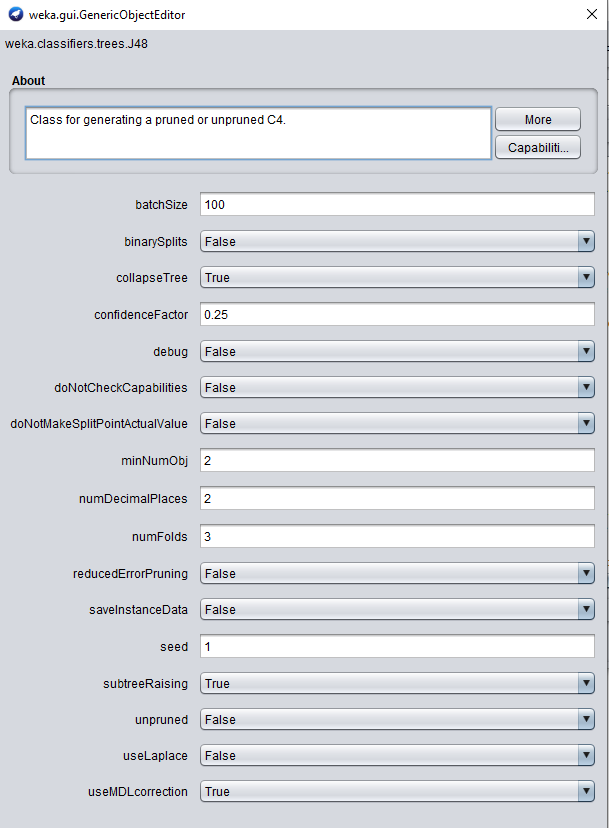
*Decision Tree Classification Model build in Weka.*

Weka version 3.9.5 was used for the project.

Preprocessed dataset DS1 was uploaded as .csv file (Please, note that DS1 and cs\_data.csv contain the same data in the same format. The only difference is cs\_data does not have the name of the attributes). In order to consider all attributes, including class label, as nominal and not numeric (which Weka recognized initially), the attribute filter “Numeric to Nominal” was applied (please, see Appendix A).

For the classification model J48 classifier was chosen - this is an algorithm to generate a decision tree that is generated by C4.5 (an extension of ID3). It is also known as a statistical classifier and one of the most popular among Weka users.

The next parameters were chosen for the classifier:



For the Test Options Percentage Split 66% was set.

For the second part of the Project – SolarWinds label prediction – the separate file containing only SolarWinds 2020 attack’s characteristics were uploaded and evaluated by the created J48 classifier.

***Output and Results Interpretation***

The output for the Java model is presented in Appendix B.

It correctly classified 16 out of 38 instances, i.e., 42%. The label “State” was 100% incorrectly classified; and, in opposite, the label “N/A” was 100% correctly classified.

Initially, the model was built in such way that it trained on 100% instances of the dataset and tested them. In that case, 109 out of 112 instances were correctly classified (accuracy 97%). That proves that model trained on a full dataset is more generalized. That was the reason to update the model by split dataset to 66% - 34%.

For the SolarWinds 2020 attack the decision tree algorithm classified it as “N/A”, the original label is “State” (based on available information).

The output of a Weka’s J48 Decision Tree is given in the Appendixes C and D.

The accuracy of the model is 65.79%.

The Weka model 100% correctly classified “N/A” class, 66.7% accurate for “Unknown”, 61.5% for “State”, and only 14.3% of “Criminal” class labels were correctly classified.

Size of the resulting Decision Tree is 27, depth is 7, and number of leaves is 14.

The Supply Chain Potential was chosen as a root by the algorithm.

As with Java model, the Weka model was initially trained on a full dataset. In that case, the accuracy was 81.25%.

For the SolarWinds 2020 attack: the model incorrectly classified it as “N/A”. The same result as with the Java model.

The reason for this SolarWinds’ misclassification could be that the attack is novel, some techniques and tactics are new, but at the same time they were used previously with other attacks but never combined into a one single attack vector. This also shows a general problem with an attack attribution in Cyber space. State actors, criminals, hacktivists, and other adversaries use same tactics and learn from each other and mimic other attackers’ tactics, tools, procedures, thus make attribution on technical side extremely difficult. Security experts should know not only tactics of different actors and their tools but be able navigate in geopolitics and criminal undergrounds.

The next table summarizes results of these two models:

|  |  |  |
| --- | --- | --- |
|  | Java | Weka |
| Correctly classified instances - 100% dataset trained | 97% | 81.25% |
| Correctly classified instances - 66% / 34% split | 42% | 65.79% |
| SolarWinds label prediction | N/A | N/A |

***Conclusion***

In the absence of publicly available research papers related to applying classification techniques to supply chain attacks datasets, it is impossible to compare the results of the project with others. The best option for the project was to build two classification - decision tree - models based on Java algorithm and Weka application. In general, Weka gives the better result with classification of data points (if not consider generalized full dataset training case). The interesting finding is that both models incorrectly classified SolarWinds attack as “N/A” instead of “State” which is widely acceptable currently in the Cybersecurity world. There are couple of reasons for low level of correct classifications. First, the size of the dataset is relatively small. Second, it is difficult to distinguish between “criminal” and “state” attacks as both have almost the same characteristics. But these left an opportunity to enhance models’ abilities to classify and predict - more publicly available data is needed and additional descriptions should be given for the attacks.

***References***

1 <https://csrc.nist.gov/glossary/term/supply_chain_attack>

2 <https://www.securityweek.com/supply-chain-attacks-nearly-doubled-2018-symantec>

3 <https://www.sonatype.com/campaign/wp-2020-state-of-the-software-supply-chain-report>

4 <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>

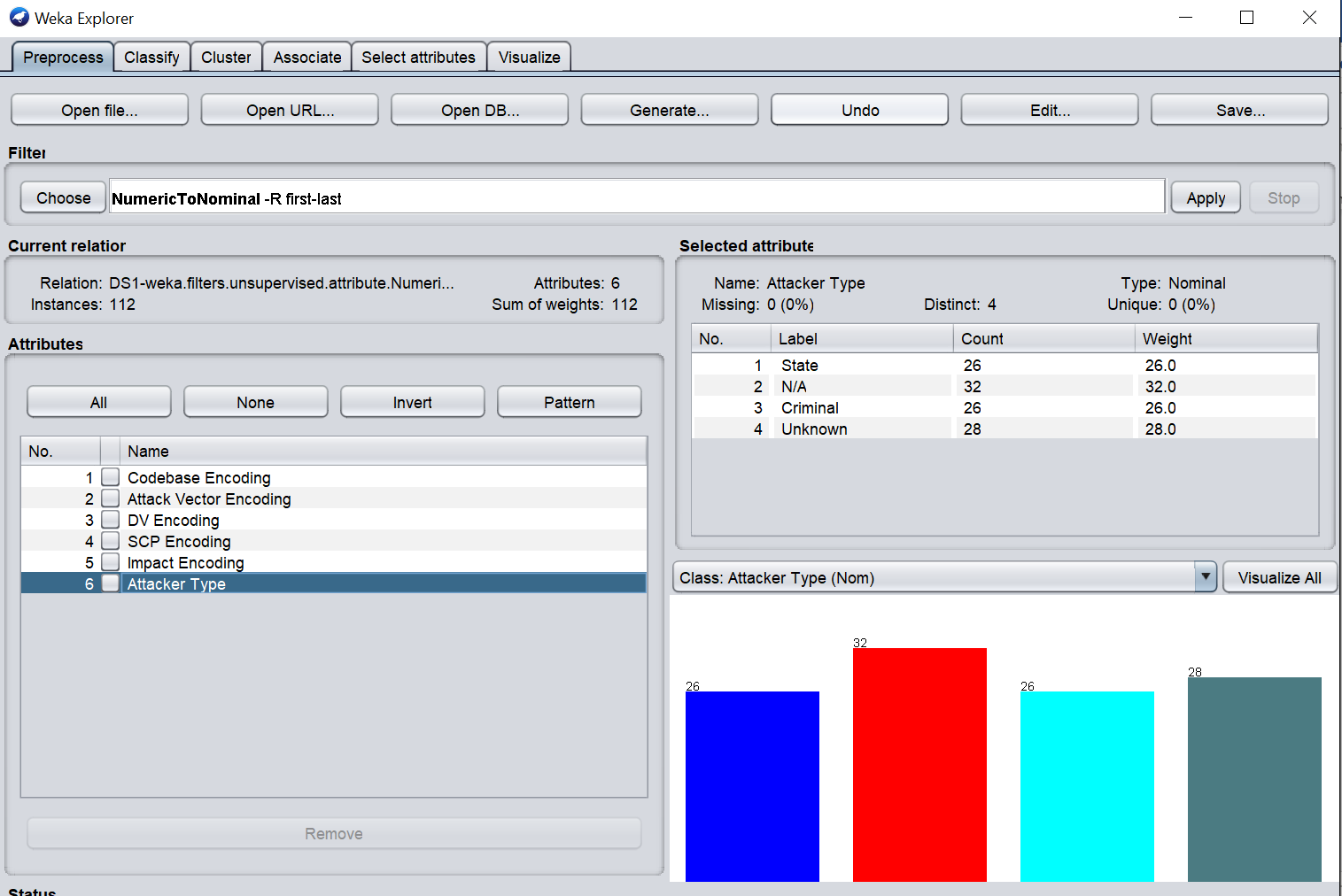
5 <https://www.mitre.org/publications/technical-papers/supply-chain-attack-framework-and-attack-patterns>

6 <https://www.researchgate.net/publication/334736415_Managing_cyber_risk_in_supply_chains_A_review_and_research_agenda>

7 Data Mining: Concepts and Techniques, Third Edition, by Jiawei Han, Micheline Kamber and Jian Pei.

**Appendix A**

**Weka - Preprocess – The Attacker Type is a Class Label**



**Appendix B**

**Java Decision Tree Model Output**

DecisionFeature: Threshold: maxInfoGain

3, 5.500000, 0.550637

4, 1010050000.000000, 0.203769

4, 5.500000, 0.158304

0, 55000.000000, 0.636514

1, 550000.500000, 0.105721

2, 5500000050.000000, 0.222305

2, 50000005.000000, 0.194976

0, 5500.000000, 0.132304

0, 1050000.000000, 0.223144

0, 505000.000000, 0.084950

1, 50500.500000, 0.174416

4, 1000050000.000000, 0.693147

1, 5.500000, 0.346574

4, 500050050.000000, 0.693147

1, 555.000000, 0.693147

2, 10050000050.000000, 0.376770

2, 5.500000, 0.212074

4, 1000000000.500000, 0.219512

2, 10000000050.000000, 0.693147

0, 1005000.000000, 0.157548

2, 10050550005.000000, 0.206192

1, 550.000000, 0.083979

1, 55.000000, 0.317535

2, 50000005.000000, 0.636514

0, 1055000.000000, 0.693147

3, 1000050.000000, 0.064719

4, 505050000.000000, 0.500402

Prediction: Actual Label

Criminal: Criminal

Criminal: Unknown

Criminal: Criminal

Criminal: Criminal

Criminal: State

Criminal: State

Criminal: Unknown

Criminal: Criminal

Criminal: State

Criminal: Criminal

Criminal: Criminal

Unknown: Criminal

State: Unknown

State: Criminal

State: Unknown

State: Unknown

State: Unknown

State: Unknown

N/A: N/A

Unknown: Unknown

N/A: N/A

State: Unknown

N/A: N/A

Unknown: Unknown

Unknown: State

Criminal: Unknown

State: Criminal

Unknown: Criminal

Criminal: Unknown

Criminal: Criminal

Criminal: Unknown

Criminal: Criminal

Criminal: Unknown

Criminal: Unknown

Criminal: Criminal

Criminal: Criminal

N/A: N/A

Unknown: State

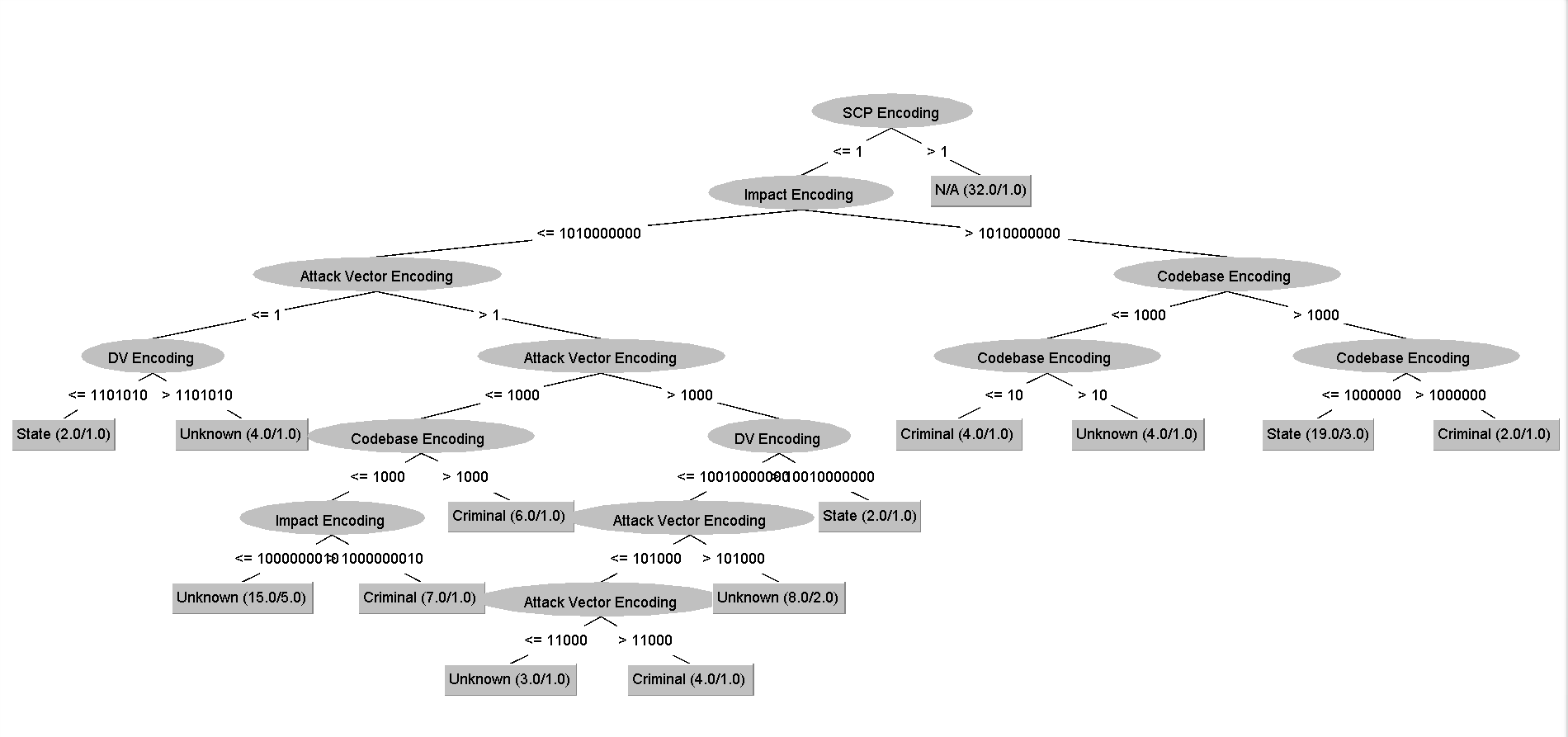
SolarWinds

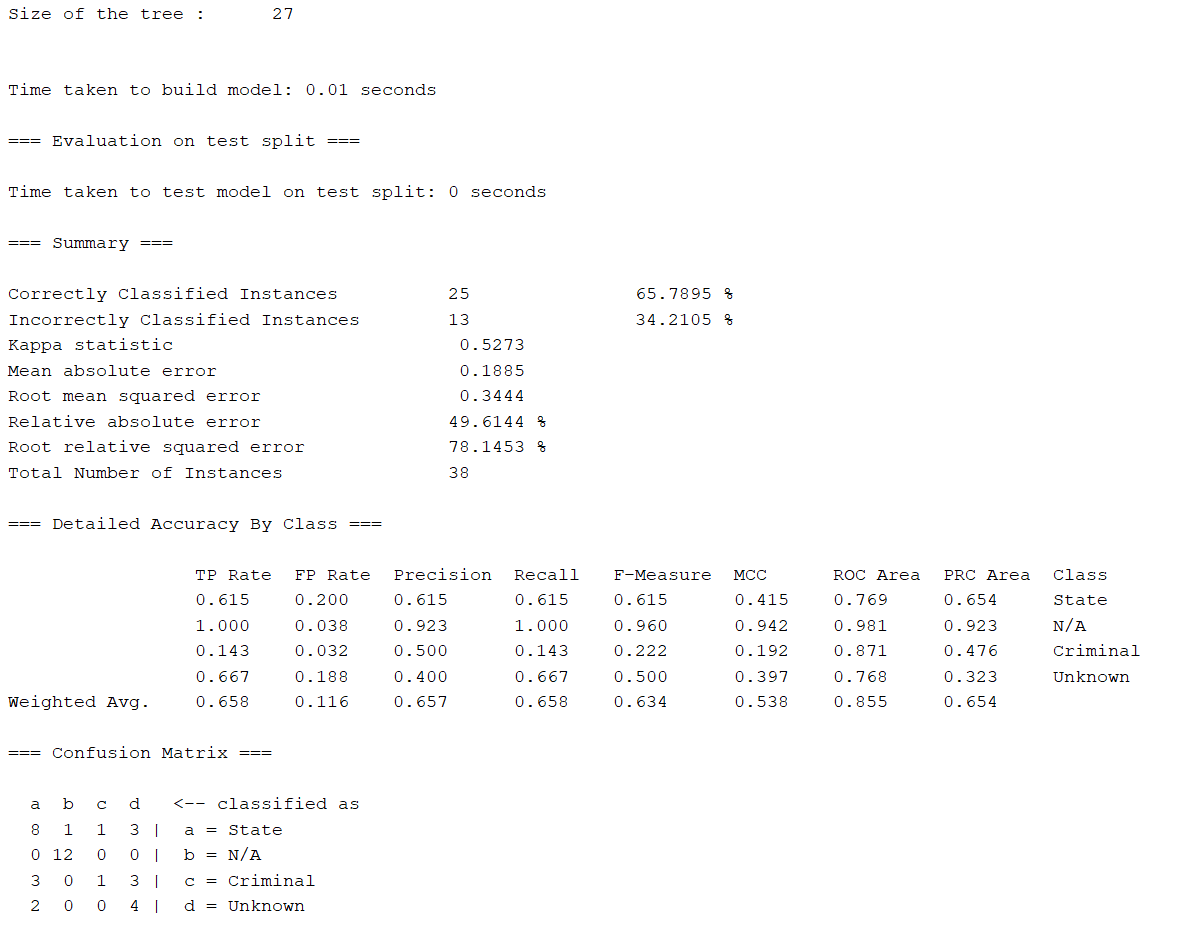
Prediction: Actual Label

N/A: State

**Appendix C**

**Weka Decision Tree Output**





**Appendix D**

**Weka Output for the SolarWinds 2020 attack**

