CSE6242 - Team NID (135) - Project Proposal Network Intrusion Detection

Akshay .
Georgia Tech
xadahiya@gatech.edu

Blaine Buxton
Georgia Tech
bbuxton6@gatech.edu

Brent Fosdick
Georgia Tech
bfosdick3@gatech.edu

Joanna Hou
Georgia Tech
chou44@gatech.edu

Jeremy Lee
Georgia Tech
jlee610@gatech.edu

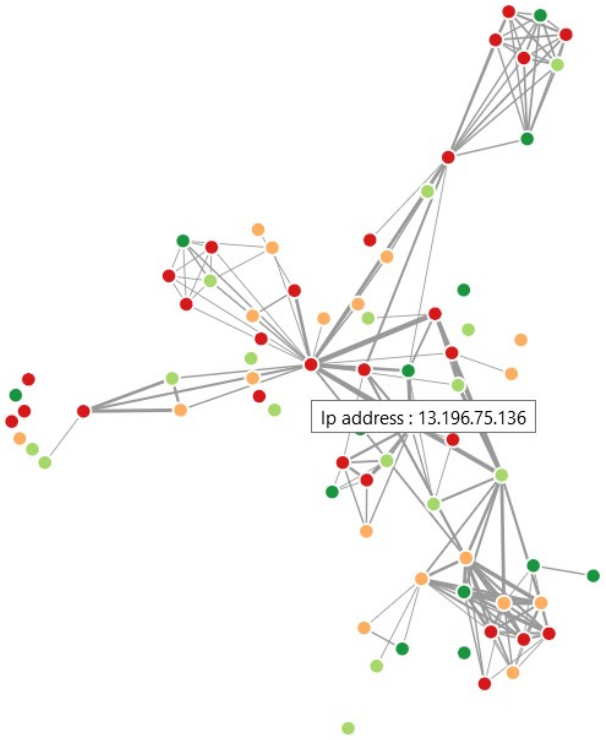

Figure 1: Prototype network displaying multiple intruders

1 OBJECTIVE

The goal of this project is to detect and visualize attacks on a computer network. This has been a hot topic of research because of the high cost of compromised assets. The problem is detecting intruders quickly while processing massive amounts of network traffic data. It has been aggravated not only by the explosion of devices like cell phones and sensor devices, but by smarter intruders who

are using automation to infiltrate. The solution has to address volume and speed while not marking legitimate traffic as malignant or attacker traffic as normal.

2 CURRENT PRACTICE

The current solutions only deal with algorithms and not visualizing what is happening in the network. The algorithms currently used center around machine learning, but not one approach is ubiquitous. The focus is to quickly classify massive amounts of network traffic data as benign or malignant. The results are then presented to network administrators and leaving it up to them as to what to do after the attack is under way.

The tool of choice to exchange data is CSV and PCAP files. The PCAP files are captured by Wireshark[4] and can be converted into CSV via command line tool tshark. This project will need to be able to process data in both formats.

Most of the solutions researched used machine learning for classification. Unsupervised machine learning techniques such as k-means[21] and support vector machines[10] have given good results with varying degrees of false positive rates. Supervised techniques such as neural nets and decision trees have also been used. The best results used aggregates of supervised algorithms for classifiers. For example, a fuzzy algorithm[15] was used for combining classifiers based on confidence. Others used more commonly accepted aggregates like random forest[13] and adaboost[1].

Some research was successful with how the data was prepared before classification. For example, one had success with "behavioral distance" [3] which calculated the distance between normal packets and intruder ones to use as a confidence score. A technique that will be explored for this project as well as other data preparation techniques.

A few surveys[7][9] were researched that compared various machine learning approaches. For an example, see Figure 2 for false positive versus true positive rates for a handful of unsupervised and

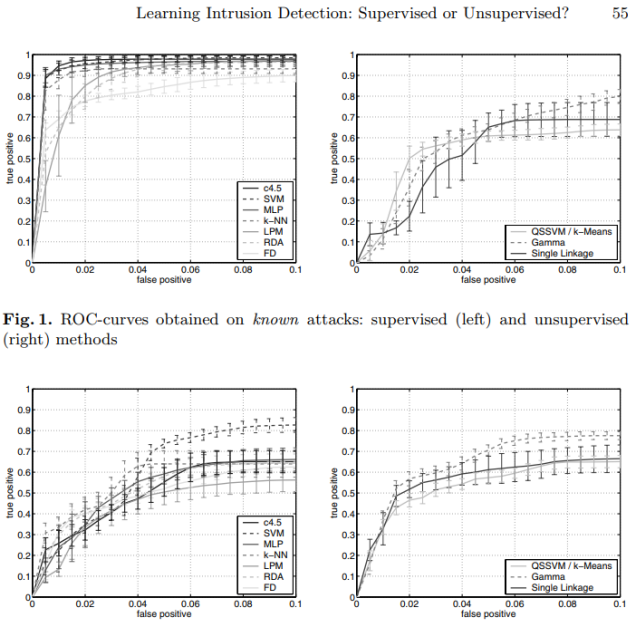

Fig. 1. ROC-curves obtained on *known* attacks: supervised (left) and unsupervised (right) methods

Fig. 2. ROC-curves obtained on *unknown* attacks: supervised (left) and unsupervised (right) methods

Figure 2: ROC graphs for Supervised vs Unsupervised ML[9]

supervised machine learning classifiers. This information will help with the research for deciding what classifiers to use. There is a trade-off between false positive rate and supervised/unsupervised techniques. Unsupervised learning detects unknown future intrusions with higher false positive rates. Supervised learning does the opposite. Even the manual for a current system[5] by IBM was researched to get a complete view of best practices.

Other novel approaches were found that used rule engines[17], pattern matching[16], and probability[20] for network detection, but might have diminishing success for this project because the inability to handle new attacks.

Most machine learning algorithms process data all at once, but this is unacceptable for detecting intruders and new attacks. New research is looking at how to make stream-based machine learning algorithms [2] to deal with the volume and keep false positives to a minimum. For example, to decrease time to process, packet headers are used[18].

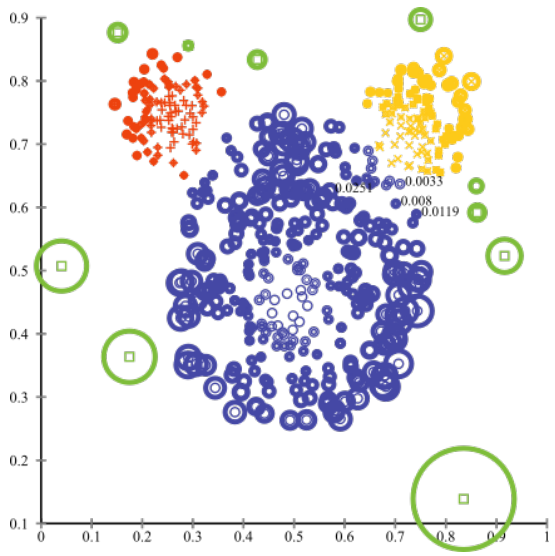

Figure 3: Example classifications using ELKI[11]

3 PROPOSED APPROACH

While machine learning techniques will be used to detect intruders, our approach is different in its emphasis on prioritizing network streams based on asset value. Visualization of the network to quickly filter the entire network to be digestible for isolating an attack. The mix of automation and visualization will allow stakeholders to more rapidly deal with intrusions. Ease of use is the ultimate goal.

The dataset, UNSW-NB15[12], will be used for training and testing of network intrusion machine learning algorithms. The KDD set [21][9] was considered, but the current art suggests to use raw TCP/IP packets[10]. The reason is that KDD has feature selection already summarized and thus, bias that could increase the false positive error rate. In addition, it will be difficult to stream the KDD.

The classifier for network intrusion will follow these principles for stream/flow based processing[8]:

- Learn incrementally.
- Only a single pass of the data.
- Perform in limited time
- Perform in limited memory

Further research will be used to learn about intruder detection [14] to test our final approach[19]. There is even a survey of visualizations that administrators expect to see[6]. The goal is to learn how an intruder thinks and what visualizations are needed to combat that. This project will go further by automating the known intrusions and making unknown intrusions easier to detect.

One aspect missing from the state of the art is prioritization of assets on the network. Not all nodes on a network are equal and it is valuable to process traffic on assets of higher value first. Prioritization will be key for the balance of speed, volume, and error.

4 AUDIENCE

Network administrators and security professionals will be the most interested in our proposal. By visualizing attacks and intruder detection on the network in real-time, they will be able to more quickly see the battle and what to triage in the case of a multiple front attack. Managers and small business owners will also be interested since the plan is to automate detection during the attack and thus not requiring dedicated security resources. Anyone who cares about the bottom line of their business will care. Thus, it must be easy to use and understand by non-technical users.

5 IMPACT

If we are successful, the downtime of a network will be lessened. Along with securing valuable assets that would have been otherwise compromised. The visualization alone will give key insights into attacks for network and security experts as well as seeing into how the detection algorithms are performing.

6 RISKS AND PAYOFFS

The risks are false positives or false negatives in the predictions. The first wrongly penalizes normal traffic and the later could let an attacker have more

time to compromise assets. The second risk is how to balance speed and incomplete information.

7 COST ANALYSIS

The cost of this proposal is the time of development resources, the devices to monitor, machines to analyze the data, machines to train algorithms, machines to classify the intruders, and finally the servers to serve the user interface.

8 COMPLETION TIME

The overall time of the project will be 70 days. See Figure 4 for the breakdown of tasks in that period. The biggest tasks are automating data cleaning and feature extraction along with researching user interface and machine learning algorithms.

9 SUCCESS MEASURES

The ultimate success of this project will be measured by the following:

- Correctly detecting new unforeseen intruder scheme.
- A false positive rate of less than 0.01.
- Less than 500 milliseconds detection time.
- Ability to scale processing data horizontally.
- Visualize confidence of intruders in network.

The checkpoint will be measured by the following:

- Handle streamed network data to scale horizontally.
- Ability to run algorithms on data with a false positive rate of less than 0.1.
- Visualize digestible portions of network.

10 PLAN

See Table 1 for details of the current high level plan and Figure 4 for the schedule and what has been completed. Github issues will be used to track tasks as the project progresses and an agile approach for completion. All team members have contributed equally and will continue to do so.

Category	Task	Owner
Analysis	Research articles	All
	Research large dataset	Jeremy
	Data cleaning and feature extraction	Joanna
	Research machine learning frameworks (SKlearn, Spark, Keras, etc)	Brent
	Research visualization frameworks (d3, onnx.js)	Blaine
	Finalize server stack for project	Akshay
	Finalize project UI framework	Brent
Design	Design machine learning classifiers	Akshay
	Create mock-ups for project UI	Jeremy
	Design Project UI	Joanna
	Design visualizations of intrusion detection in UI	Blaine
Implementation	Setup network infrastructure for the intrusion demo	Blaine
	Write test automation	Joanna
Documentation	Write proposal document	Blaine
	Write slides for proposal video	Brent
	Film proposal video	Brent
	Write progress report	Joanna
	Write final report	Jeremy
	Write final presentation slides	Akshay
	Film final presentation video	Jeremy
Management	Scrum master	Akshay

Table 1: Plan and Assignments

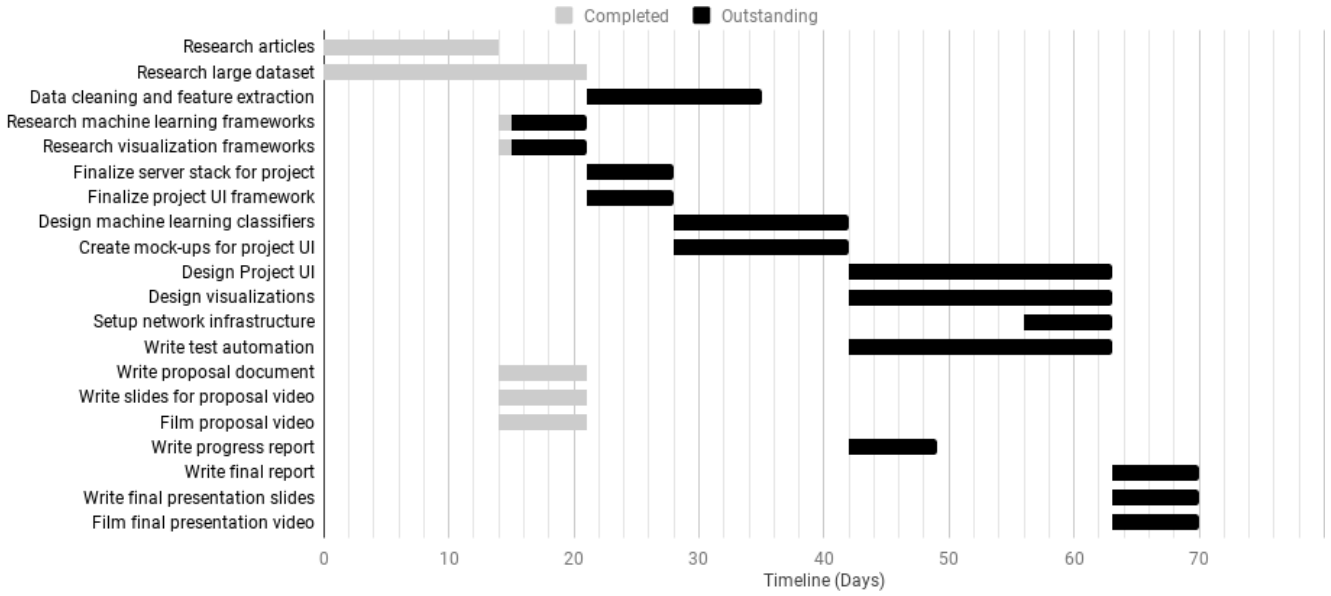

Figure 4: Plan Schedule

REFERENCES

[1] Lai Cao, Wei Han, and Sheng Dong. 2014. A New Intrusion Detection Method Based on Machine Learning in Mobile Ad Hoc NETwork. *Applied Mechanics and Materials* 548-549, Achievements in Engineering Sciences (2014), 1304–1310. http://search.proquest.com/docview/1523679531/

[2] Lennart Van Efferen and Amr M.T. Ali-Eldin. 2017. *A multi-layer perceptron approach for flow-based anomaly detection.* https://ieeexplore.ieee.org/abstract/document/8072036

[3] Debin Gao, Michael K. Reiter, and Dawn Song. 2006. Behavioral Distance for Intrusion Detection. In *Recent Advances in Intrusion Detection*, Alfonso Valdes and Diego Zamboni (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 63–81.

[4] Shilpi Gupta and Roopal Mamtora. 2012. *Intrusion Detection System Using Wireshark.* http://ijarcsse.com/Before_August_2017/docs/papers/11_November2012/Volume_2_issue_11_November2012/V2I11-0205.pdf

[5] IBM. 2013. *Security Intrusion detection.* https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzaub/rzaubpdf.pdf

[6] A. Karami. 2018. An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities. *Expert Systems with Applications* 108 (2018), 36–60.

[7] Nathan Keegan, Soo-Yeon Ji, Aastha Chaudhary, Claude Concolato, Byunggu Yu, and Dong Hyun Jeong. 2016. A survey of cloud-based network intrusion detection analysis. *Human-centric Computing and Information Sciences* 6, 1 (05 Dec 2016), 19. https://doi.org/10.1186/s13673-016-0076-z

[8] Manish Kumar and M Hanumanthappa. 2013. Intrusion detection system using stream data mining and drift detection method. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, 1–5.

[9] P Laskov, P Dussel, C Schafer, and K Rieck. 2005. Learning intrusion detection: Supervised or unsupervised? *Image Analysis And Processing - Iciap 2005, Proceedings* 3617 (2005), 50–57.

[10] P. Manandhar and Z. Aung. 2014. Towards practical anomaly-based intrusion detection by outlier mining on TCP packets. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8645, 2, 164–173.

[11] Anony Mousse. 2015. *Anomaly detection based on clustering.* https://stats.stackexchange.com/questions/160260/anomaly-detection-based-on-clustering

[12] Nour Moustafa and Jill Slay. 2016. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective* 25, 1-3 (2016), 1–14.

[13] Takeshi Okamoto. 2011. An artificial intelligence membrane to detect network intrusion. *Artificial Life and Robotics* 16, 1 (2011), 44–47.

[14] Alexander Prohorenko. 2000. *TCP hijacking.* https://www.techrepublic.com/article/tcp-hijacking

[15] Inez Raguenet and Carlos Maziero. 2008. A Fuzzy Model for the Composition of Intrusion Detectors. In *Proceedings of The Ifip Tc 11 23rd International Information Security Conference*, Sushil Jajodia, Pierangela Samarati, and Stelvio Cimato (Eds.). Springer US, Boston, MA, 237–251.

[16] M. Richard, Guan-zheng Tan, P. Ongalo, and W. Cheruiyot. 2013. Novel design concepts for network intrusion systems based on dendritic cells processes. *Journal of Central South University* 20, 8 (2013), 2175–2185.

[17] Martin Roesch. 1999. *Snort - Lightweight Intrusion Detection for Networks.* http://static.usenix.org/publications/library/proceedings/lisa99/full_papers/roesch/roesch.pdf

[18] Carol Taylor and Jim Alves-Foss. 2002. An empirical analysis of NATE: Network Analysis of Anomalous Traffic Events. In *Proceedings of the 2002 workshop on new security paradigms (NSPW '02)*. ACM, 18–26.

[19] T. Verwoerd and R. Hunt. 2002. Intrusion detection techniques and approaches. *Computer Communications* 25, 15 (2002), 1356–1365.

[20] Yun Wang and Inyoung Kim. 2008. A Bootstrap-based Simple Probability Model for Classifying Network Traffic and Detecting Network Intrusion. *Security Journal* 21, 4 (2008).

[21] Guo Yin Wang Yong Jun Hai, Yu Wu. 2005. An improved unsupervised clustering-based intrusion detection method. , 5812 - 5812 - 9 pages. https://doi.org/10.1117/12.603086