Automated Fast-flux Detection using Machine Learning and Genetic Algorithms

Sachin Rana, Dr. Ahmet Aksoy

Introduction

- In a cyber attack, attackers aim to **cover their trail**
- Fast-flux is used by **malicious bots** to hide their C2 servers
- These bots are infected by **malware**
- A network of such bots is controlled remotely by a **bot-master** [1]
- **Rapid changes** to the hosts are applied [2]
- This helps make the network more **resistant to discovery**
- It is **essential** to detect such networks
  - They may be used to send spam/phishing emails with **links to the malicious servers** [3]

Introduction (cont'd.)

- In this paper, we present an **automated** fast-flux host detection approach
- We use machine learning (**ML**) and genetic algorithms (**GA**)
- Our approach can identify fast-flux hosts from a **single packet** with high accuracy
- With the help of ML and GA, our approach automatically identifies packet header fields in TCP/IP stack **without expert input**

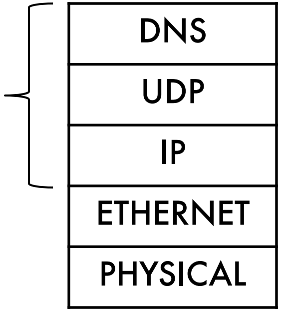
Our Approach

- Advantages:
  - Feature selection in a **reasonable amount of time** with GA
    - With a brute-force approach on 30 features, 2^{30} combinations need to be considered
  - **No hand selection** of "useful" features like DNS TTL, etc.
    - GA automatically selects features that yield as high accuracy as possible
  - No dependence on **signatures from any other tools**
    - Our tool generates signatures for malicious and benign packets by itself
  - **Insusceptible** to malicious hosts' behavior
    - By retraining on the updated dataset, our approach can automatically pick up uniqueness in behavior

Our Approach (cont'd.)

- Advantages: (cont'd.)
  - **No preset number of features** used
    - Our approach selects minimum number of features necessary to achieve as high accuracy as possible
  - To the best of our knowledge, **the first to employ** a GA to automatically detect distinguishing features to detect fast-flux without expert input
- Disadvantage:
  - **Complete–dependance on the dataset** provided
    - The data needs to be as representative as possible

Methodology



- We consider the **TCP/IP headers** for DNS packets
  - Which include lower-level IP and UDP protocols
- We tested **various ML algorithms** for their contribution
  - Decision Trees (J48), PART, Decision Table (DT), Decision Stump (DS), Artificial Neural Networks (MLP), Random Forest (RF), Support Vector Machines (SMO), JRip, Logistic Regression, and Bayesian Network
- We analyzed each of these algorithms' **classification accuracy**
- We analyzed the one that generated **the highest accuracy** in each case

Data Initialization

- **ISOT Botnet** and **CTU-13** datasets for the malicious packets
- **ISOT HTTP** dataset for the benign packets
- We filtered datasets for only **DNS response packets**
  - To analyze the DNS behaviors of hosts
- We removed features that would introduce **bias**
  - Such as IP addresses, IP identifiers, checksum (that contain IP addresses), and timing features

TABLE I: No. of Packets

Dataset	Class	*.19	*.50	*.56	*.57	*.63
Dataset 1	Malicious	5506	5506	5506	5506	5506
	Benign	142	278	39	612	1211
Dataset 2	Malicious	22734	22734	22734	22734	22734
	Benign	142	278	39	612	1211

Feature Selection

1	0	0	0	1	0	1	0	0	1	0	0	1	0

- In GA, **chromosomes** represent the solutions
- A chromosome contains **a series of 0's and 1's**
- **In our implementation**, a solution is a series of 0's and 1's
  - Which are at the same length as the number of features available in the data
- If the corresponding bit of a feature in a GA solution is **0, feature is ignored**, and if it is **1, feature is considered**
- GA runs the **fitness function** to determine a potential solution's contribution

$$Fitness = 0.98 \times Accuracy + 0.02 \times \left(1 - \frac{|SelectedFeatures| - 1}{|AllFeatures| - 1}\right)$$

Experimental Results

- We used **IP**, **UDP**, **DNS**, and **IP & UDP & DNS** features, respectively
  - To demonstrate each protocol's contribution to fast-flux detection
- Using **IP features only**
  - We observed IP Flags, IP TTL, and IP Length features were selected by GA
  - This does not necessarily indicate that the IP protocol alone is reliable for fast-flux detection
  - Such features are prioritized in performing Operating System, and IoT device fingerprinting as well

Experimental Results (cont'd.)

- Using **UDP features only**
  - We observed 99% classification accuracy using the UDP length feature alone
  - This is mainly because the UDP length includes the DNS packet's length as well
  - However, this is still not very reliable!

Experimental Results (cont'd.)

- Using **DNS features only**
  - We observed 100% classification accuracy using DNS features alone
  - GA selected at most five features for a single host at a time
  - In general, we observed the occurrence of 8 DNS features across all runs
  - Consistent with previous research, GA captures features known to help detect fast-flux
    - The number of authoritative name servers [3,12,18,19]
    - The number of additional records [3,12,18,19]
    - The length of the response packet
    - The DNS query/response type [12]
    - The DNS query name length [3,12,18,19]
    - The DNS response TTL [3,12,18,19,20]

Experimental Results (cont'd.)

- Using **IP & UDP & DNS features together**
  - We were able to classify the packets for both datasets at 99.9% accuracy
  - The features selected remained consistent with our findings from when we used DNS features alone

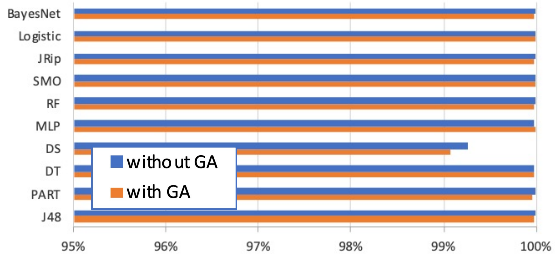


Fig. 3: Dataset 1 (without TTL)

TABLE III: Dataset 1 GA-selected Features (without TTL)

Features	*.19	*.50	*.56	*.57	*.63	Σ
dns.count.add_rr		✓	✓	✓	✓	4
dns.count.auth_rr	✓	✓		✓	✓	4
ip.len	✓	✓		✓		3
udp.length	✓		✓			2
dns.flags.authoritative		✓	✓			2
dns.flags		✓				1
dns.resp.ttl			✓			1
ip.flags			✓			1
Σ	3	5	5	3	2	

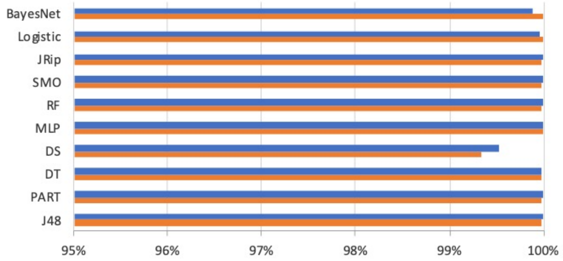


Fig. 5: Dataset 2 (without TTL)

TABLE IV: Dataset 2 GA-selected Features (without TTL)

Features	*.19	*.50	*.56	*.57	*.63	Σ
dns.count.auth_rr	✓	✓	✓	✓	✓	5
ip.len		✓		✓	✓	3
dns.count.add_rr	✓		✓			2
dns.qry.name.len	✓		✓			2
dns.qry.type		✓				1
dns.resp.ttl				✓		1
udp.length					✓	1
Σ	3	3	3	3	3	

Conclusion

- We presented a **completely automated single-packet** fast-flux detection using ML and GA
- Our approach automatically selects a subset of features that **contribute most** to the classification of benign and malicious packets
- Feature selection also helps eliminate **features that cause noise**
  - In some cases, using GA yielded **higher accuracy** than when all the features were used together

Conclusion (cont'd.)

- Our approach achieved more than 99% classification accuracy **using less than half of the features** in DNS packet headers
- GA-selected features with no expert input were **highly consistent** with the features used in fast-flux detection
- If malicious systems' **behavior changes** to evade detection, retraining on the updated dataset is expected to capture the new behavior

Future Work

- We would like to consider the **statistical values** of each feature in the dataset
  - Such as the minimum/maximum number of IPs in DNS response packets and the average TTL
- This would help our approach become even more **insusceptible to the possible changes** in malicious hosts' behavior

Thank you!