On Scalable Attack Detection in the Network

Ramana Kompella,
Sumeet Singh,
George Varghese

UCSD
Motivation

- **Impact:** Attacks such as Worms, DoS => billions of dollars in damage
- **Collateral Damage:** Detecting attacks early in network reduces collateral damage
- **Scalable Detection:** High speed attack detection in the network requires scalability in state
- **This Talk:** Issues and algorithms for scalable attack detection using DoS as an example.
Collateral Damage

Detecting and Defending in the network reduces collateral damage.
Existing Approaches

- **End host** based approaches (e.g., Syn-cookies) can avoid problem but
  - Deployment: Need to patch all end-nodes
  - Network bandwidth: wasted

- **Perimeter** Approaches (e.g., Riverhead, Michigan) detect and discriminate spoofed sources via TTLs but:
  - Zombies: TTL based approaches cannot handle DDoS attacks via non-spoofed sources such as zombies
  - Network Bandwidth: Wasted (upto point of detection)

- **Other scalable approaches**
  - Multistage filters, sketches, MULTOPS
  - Traditionally only bandwidth based attacks
Challenges with attack detection in the network

- Main Challenge: Scalability
  - Large number of flows, hence per-flow state difficult
  - High speed links, hence less processing time

- Why Scalability in State is Important
  - High speed memories (SRAM) don’t scale in density as well as lower speed memory (DRAM)
  - Even if available, high speed memory is expensive

How to scalably detect attacks in the network?
Scalability via Aggregation

- Analogies
  - IP addresses are aggregated using prefixes
  - Flows are aggregated into DiffServ Classes

- Key Insight
  - Maintain state across aggregates (instead of per-flow)
  - Example: Subnet aggregation, Hash buckets

- But Aggregation introduces issues:
  - Behavioral Aliasing
    - Good behaviors aggregate to look bad (False Positives)
    - Bad behaviors might look good (False Negatives)
  - Spoofing: More details in the paper
    - Attack behavior can be spoofed to look benign
Contributions

- Framework: Any scalable attack detection algorithm must address behavioral aliasing & spoofing.
- Concrete technique: Data structure called PCF (Partial Completion Filter)
- Evaluation of PCFs:
  - Behavioral Aliasing using Gaussian approximation
  - Spoofability
Attack Classes we Address

- Partial Completion Attacks
  - TCP SYN Flooding, Naptha attacks

- Attacks that use scanning
  - Portscans, worm scanning, backdoor probes

- Much existing work on these problems, but little focus on scalability.
Naïve Solution

- Simple solution
  - Maintain state (count of number of SYN packets - count of number of FIN packets)
  - Requires per-flow state
  - Not feasible in the network

- We use a Partial Completion Filter to
  - Identify a behavior: Identify flows (destinations) that have a high imbalance in SYN and FIN packets in a given interval
  - Perform Aggregation: We use randomized hashing (unbiased) to aggregate flows
Partial Completion Filter: Idea

**Subset of Packets mapped To Bucket 2**

```
SYN
FIN
SYN
SYN
SYN
SYN
```

**Attack Stream**

```
SYN
SYN
SYN
SYN
SYN
SYN
```

**Benign Traffic**

**Subset of Packets mapped To Bucket 6**

```
SYN
FIN
SYN
FIN
SYN
SYN
```

**Hash Buckets**

```
PCF | 4 | 1 |
----|----|--
```
Behavioral Aliasing: Modeling Benign Traffic

- Long-lived connections
- Retransmissions
- Route churn

Benign behavior
Prob (Packet = SYN) = 0.5
Prob (Packet = FIN) = 0.5
Behavioral Aliasing: Modeling Benign behavior

Benign behavior
Prob \((X_i = 1) = 0.5\)
Prob \((X_i = -1) = 0.5\)
Mean \((X_i) = 0\)
Variance \((X_i) = 1\)

What is the distribution of \(X = \sum X_i\) ?
Behavioral Aliasing: Gaussian Approximation

\[ X = \sum X_i \] can be approximated using Gaussian

Hash Buckets
Additive Noise

- Noise due to benign connections is additive with the attack flow size
- If attack is SYN-FIN difference > T, then
  - Choose a threshold T-3sigma
  - PCF identifies any flow $f > T$ w.h.p (1 - False Negative Probability)
  - PCF identifies a flow $f$, then $f > T-6sigma$ w.h.p (1 - False Positive Probability)
Behavioral Aliasing: False Positives

FALSE POSITIVE due to mapping to attack flows

\[ p = \text{probability that a benign flow maps to an attack flow’s bucket} \]

\[ p^k = \text{probability that a benign flow maps to an attack flow in all k stages} \]
Partial Completion Filter with multiple stages

- **Field Extraction**
- Hash 1
- Counter
- Stage 1
- Comparator
- Hash 2
- Counter
- Stage 2
- Comparator
- Hash 3
- Counter
- Stage 3
- Comparator

**Alert!**
If all counters above threshold

Increment for SYN, Decrement for FIN
Reducing False Positives Further

- Add multiple stages with independent hash functions
- More stages reduces false positives exponentially
  - A flow is selected only if buckets in all stages greater than a threshold
- But false negatives increases linearly (approx.)
- So, choose number of stages so that false positives reduced drastically while false negatives don’t increase too much
Measurement on Real Traces

- Traces from two different ISPs A and B (CAIDA)
  - Named as ISPA Dir-0, Dir-1, ISPB Dir-0, Dir-1

- Tuning the Parameters
  - How to choose bucket sizes?
    - We use 5000 buckets
  - How big should the measurement interval be?
    - Small: fast detection but lacks clear signature to infer an attack.
    - Large: increased time of detection
    - We use 1 minute as the measurement interval
How does SYN-FIN asymmetry affect?

Bias towards SYNs cause slight shift in the noise.
How many stages are good enough?

Tradeoff between False Positives and Negatives
Deployment

- Syn Flood
  - Forward Path of the attack traffic: PCF (SYN, FIN, <DIP, DP>)
  - Reverse Path of the attack traffic: PCF (SYN, FIN, <SIP, SP>)

- Port Scanning, Worm Scanning
  - PCF (SYN, FIN, <SIP>)

- Horizontal Port Scan
  - PCF (SYN, FIN, <SIP, DP>)

- Bidirectional deployment
  - PCF (outgoing SYN, incoming FIN, <IP, Port>)

- More details about spoofability in different deployment options in the paper
Experience of PCFs over large timescales (1 day)

- Total Destinations = 5.16 Million
- Total <DIP,DP> pairs = 30.36 Million
- Threshold = 150 (sigma=10)
  - 2.5 SYN-FIN imbalances per second (avg)
- Total flows detected = 517
- False Positives = 6
- False Negatives = 0

More about actual characteristics of these flows such as duration and so on in the paper
Conclusions

- **Scalable attack detection framework**
  - Behavioral aliasing and spoofing are key questions for any scalable detection technique

- **PCF Data structure**
  - Provides a general tool to detect wide range of partial completion attacks

- **Analysis**
  - Gaussian Approximation to estimate behavioral aliasing.

- **Evaluation**
  - Validated against real network traces

- **Low Cost Implementation**
  - Using just 360KB of High speed memory