Stony Brook University

IARIA

MAVERICK:
Detecting Network Configuration & Control Plane Bugs Through "Structural Outlierness"

Vasudevan Nagendra

Plume Design Inc

vnagendra@plume.com, vnagendra@cs.stonybrook.edu

Abhishek Pokala*
Stony Brook University

Arani Bhattacharya*
IIIT Delhi

Samir R Das
Stony Brook University

Wireless Networking and Systems Lab (WINGS)
Stony Brook University

* Work done when at WINGS Labs, Stony Brook University

SECURWARE 21 Athens, Greece

Bio

- Head of Security @ Plume Design Inc
  - Consumer Security (Smart Homes, Small businesses)
  - Over 40 millions homes/businesses (1.2 Billion devices)
- Over 15+ years of experience building Network Security productions & solutions

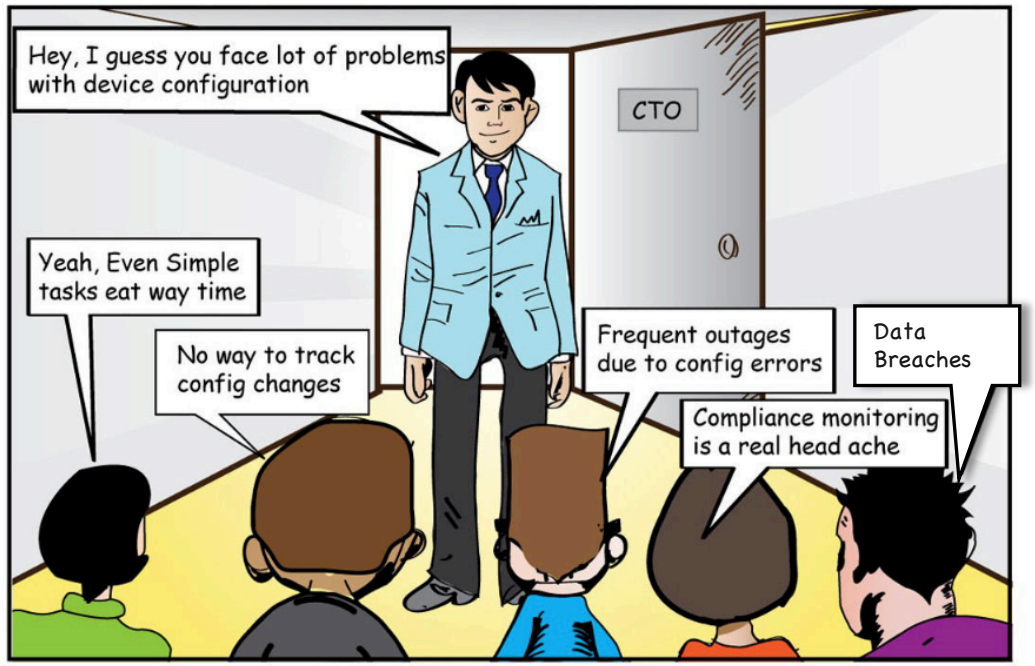
Active research

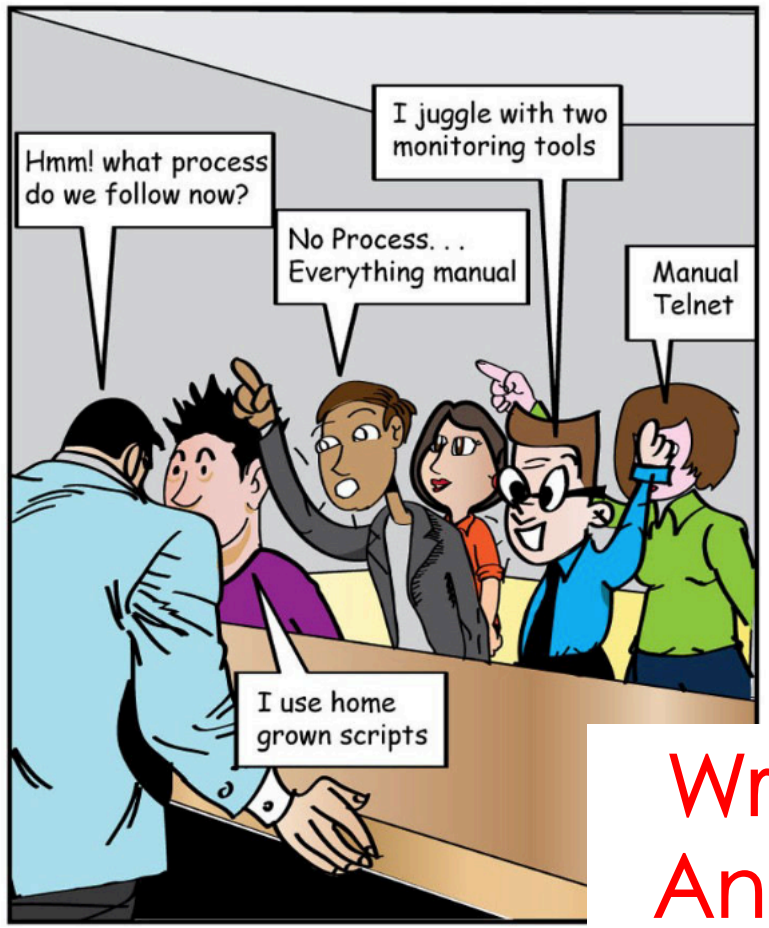
- ML-based Anomaly Detection for zero-days
- Context-aware IoT security
- Privacy-enabled technology for security platforms
- Advanced data planes (monitoring/enforcement)
  - For zero-days and vulnerabilities
- Device and application typing and fingerprinting
- IoT Behavioral Analysis
- Client deduping tech for Mac Randomization
- Network Verification
  - Cloud, Enterprise

Problem: How to handle network configurations

Question: Network Manager / NetOps / CIO / CISO / CTO

ManageEngine - Network Configuration Manager

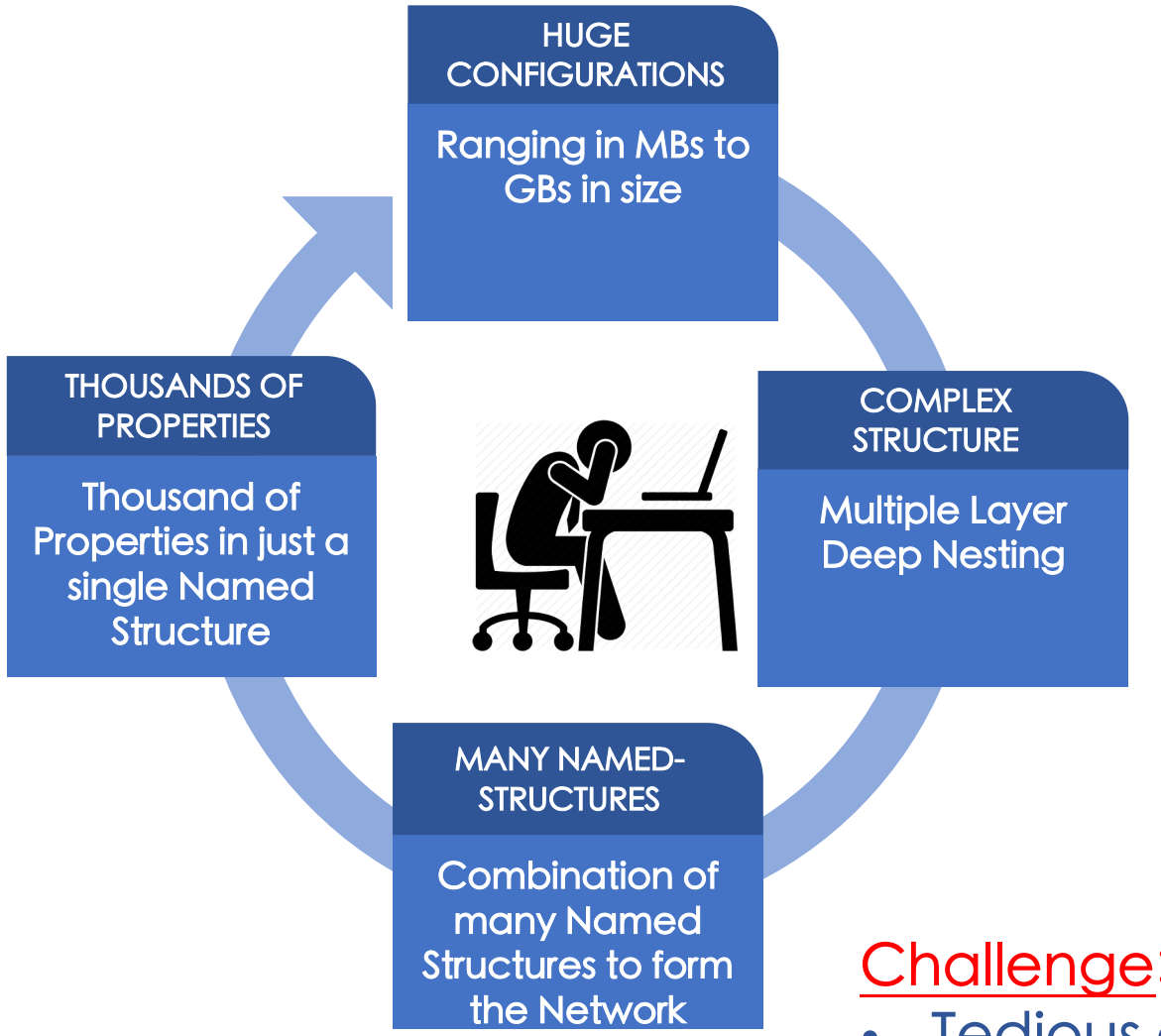


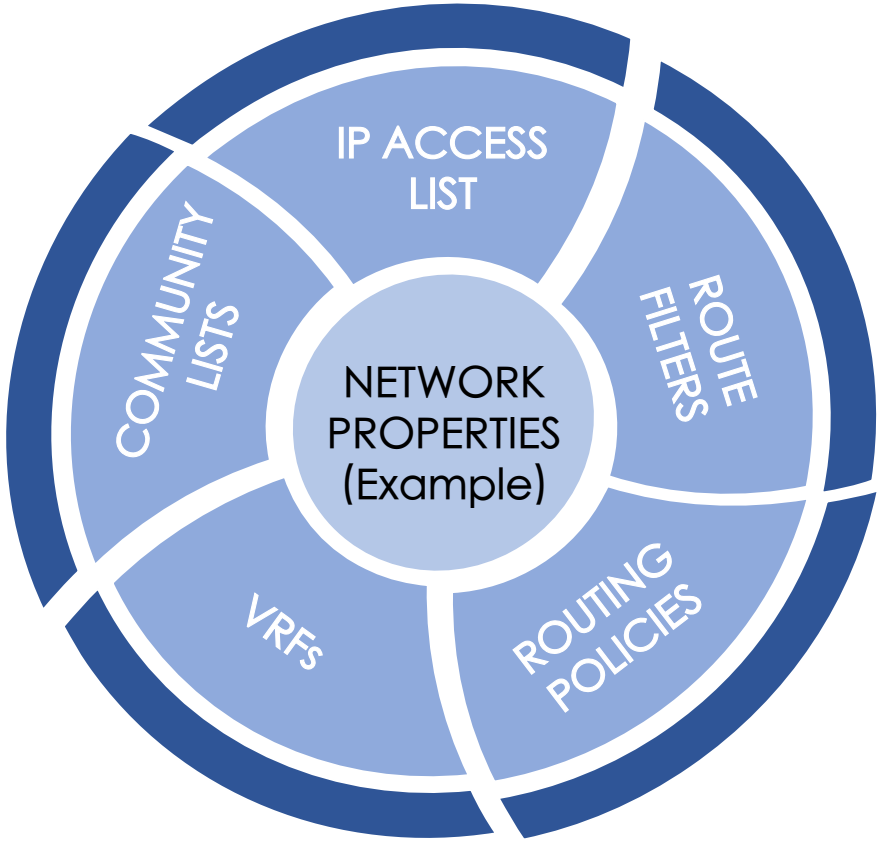


Wrong Answer

Credit: https://download.manageengine.com/network-configuration-manager/Network-configuration-management-webcomic.pdf

Challenges: Handling network configurations





Hundreds of different types of Network properties …

Challenge:

- Tedious and prone to errors and bugs

Challenge: Heterogeneity & volume

Multiple Vendors

Cisco Juniper Arista Aruba
AWS F5
Dell Force 10 Palo Alto Networks

Heterogeneous configuration format

Firewalls Switches Application
Edge Routers Hosts Gateways
Virtual Network Functions

Complexity:

- Millions of lines of configurations
- Vendor-specific syntax and configuration formats !!!

Challenge:

- Automatically Finding Bugs in Network Control Plane
  - Minimal or no manual intervention
- Detecting bugs with least number of FPs and mostly 0 FNs

Result: Existing network verification or lack of it

1. Network outage
2. Data loss / leaks
3. Network / data breach and compromise

Problem: Hundreds of outages costing billions $

Newsweek

Newsweek

Newsweek

October 2021: Facebook Outage

June 2021: Fastly Outage

Newsweek

ovember 2020: Amazon Web Services

he Amazon Web Services (AWS) logo, a division of Amazon.com's US e-commerce group is displayed during the 4th edition of he Viva Technology show at Parc des Expositions Porte de Versailles on May 17, 2019 in Paris, France.
HESNOT/GETTY IMAGES

Newsweek

Newsweek

August 2020: CenturyLink/Level 3

Close-up of logo on facade at headquarters of cyber security company Cloudflare in the South of Market (SoMA) neighborhood of San Francisco, California, June 10, 2019.
SMITH COLLECTION/GADO/GETTY IMAGES

July 2021: Bug at Akamai

An attendee walks past the PlayStation logo in the Sony Interactive Entertainment booth during the Tokyo Game Show 2018 on September 20, 2018 in Chiba, Japan. The PlayStation Network was among those affected by the Akamai outage.
TOMOHIRO OHSUMI/GETTY IMAGES

On July 22, sites like Amazon, UPS, Airnbnb, the PlayStation Network, Steam and FedEx went down, all thanks to an outage with the Akamai Edge domain name system (DNS) service.

People trying to access these sites, plus others including American Express, Delta Airlines and Home Depot, were met with a DNS error message.

Cause for major Network outages:

Network/Service configuration errors by human

Credit: https://www.newsweek.com/major-website-outages-that-wreaked-havoc-2020-2021-1640031

Industry Surveys: Cost of each downtime?

"*What is the cost of network downtime?*" Based on industry surveys, the number we typically cite is $5,600 p/minute, which extrapolates to well over $300K p/hour (see Ensure Cost Balances Out With Risk in High-Availability Data Centers by Dave Cappuccio).

Based on the characteristics of your business and environment (i.e., your vertical, risk tolerance etc). For example, this Avaya study indicates the range is from $140K to $540K p/hour.

Credit: https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/

Problem: Breaches, data leaks, & compromises

Cloud / Enterprise security misconfiguration

CYPRESS: In 2017 cloud security misconfiguration accounted for almost 70% of the overall compromised data records

CYPRESS: In 2018 Exactis breach, where 340 million records were exposed, affecting more than 21 million companies.

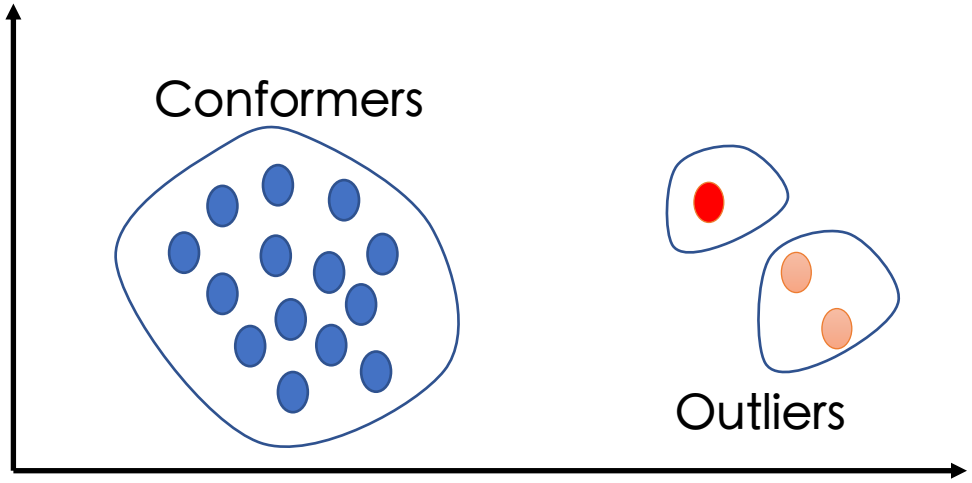
PURPLESEC: In 2018 Configuration errors accounted for ~52% of the attacks and data breaches in cloud and enterprise networks

ThreatStack: In 2020 73% of organizations have at least one critical security misconfiguration that could expose critical data and systems or enable attackers to gain access to sensitive information or private services or to the main AWS (Amazon Web Services) console.

Credit: https://www.guardicore.com/blog/understanding-and-avoiding-security-misconfiguration/
https://www.threatstack.com/blog/73-of-companies-have-critical-aws-security-misconfigurations

Outliers means to verify network configurations

An *outlier* is an observation that is abnormal from other values in the population



Outliers in Network Configurations:

- Absence of a network property
- Deviation from actual definition

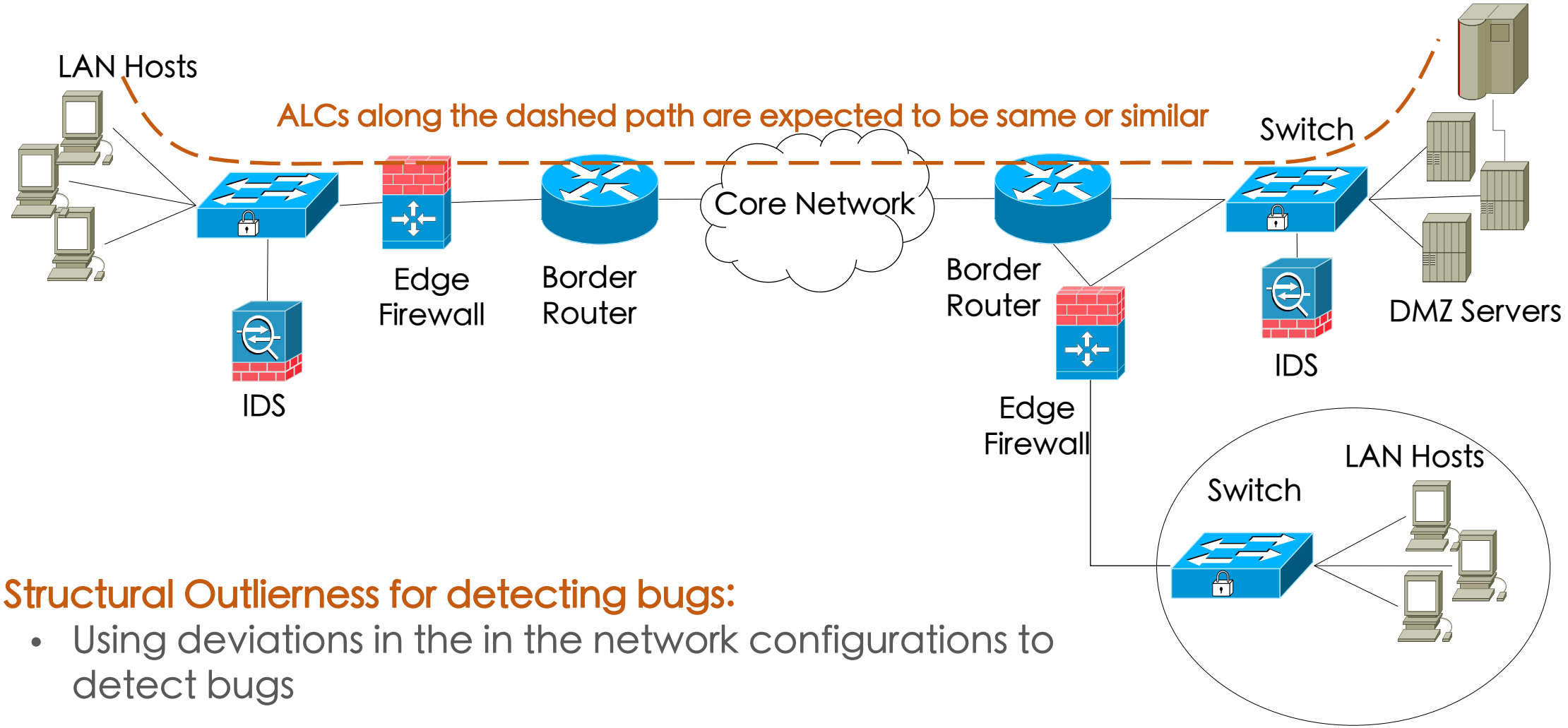
Existing Approaches: Outlier detection mechanisms

- Statistical Methods
  - Data assumed to follow a distribution model (e.g., Normal distribution)
  - Outlier: Significant deviation from the distribution
  - Issues: High mis-classification rate, flagging intentional config changes

- Proximity-based Model
  - Partition of data and measure of fit of data points to different partitions.
  - No assumption about data model
  - Outlier: Distance of data to different partitions
  - Issues: High mis-classification rate

- Logical or Rule-based Approach
  - Building rules and regex out of the configurations
  - Outlier: Exact matching with the rule or regex for outlier detection
  - Issues: Requires low-level vendor-specific rules, tedious, prone to error, Misses bugs

MAVERICK: Our approach



- Structural Outlierness for detecting bugs:
  - Using deviations in the in the network configurations to detect bugs

MAVERICK: Approach steps

Simple Example configurations (ACLs)

ACL-1: {action:PERMIT, matchCondition:{class: temp1, headerSpace: {ipProtocols: ['TCP']}}, srcPorts: ['51102-51102']}
ACL-2: {action:PERMIT, matchCondition:{class: temp1, headerSpace: {ipProtocols: ['TCP']}}, srcPorts: ['51102-51102']}
ACL-3: {action:PERMIT, matchCondition:{class: temp2, headerSpace: {ipProtocols: ['TCP']}}, srcPorts: ['51102-51103']}
ACL-4: {action:PERMIT, matchCondition:{class: temp1, headerSpace: {ipProtocols: ['TCP']}, srcPorts: ['31002-51104'], }

Clustering and separating outliers:
[Cluster 1: [ACL-1, ACL-2, ACL-3], [ACL-4]]

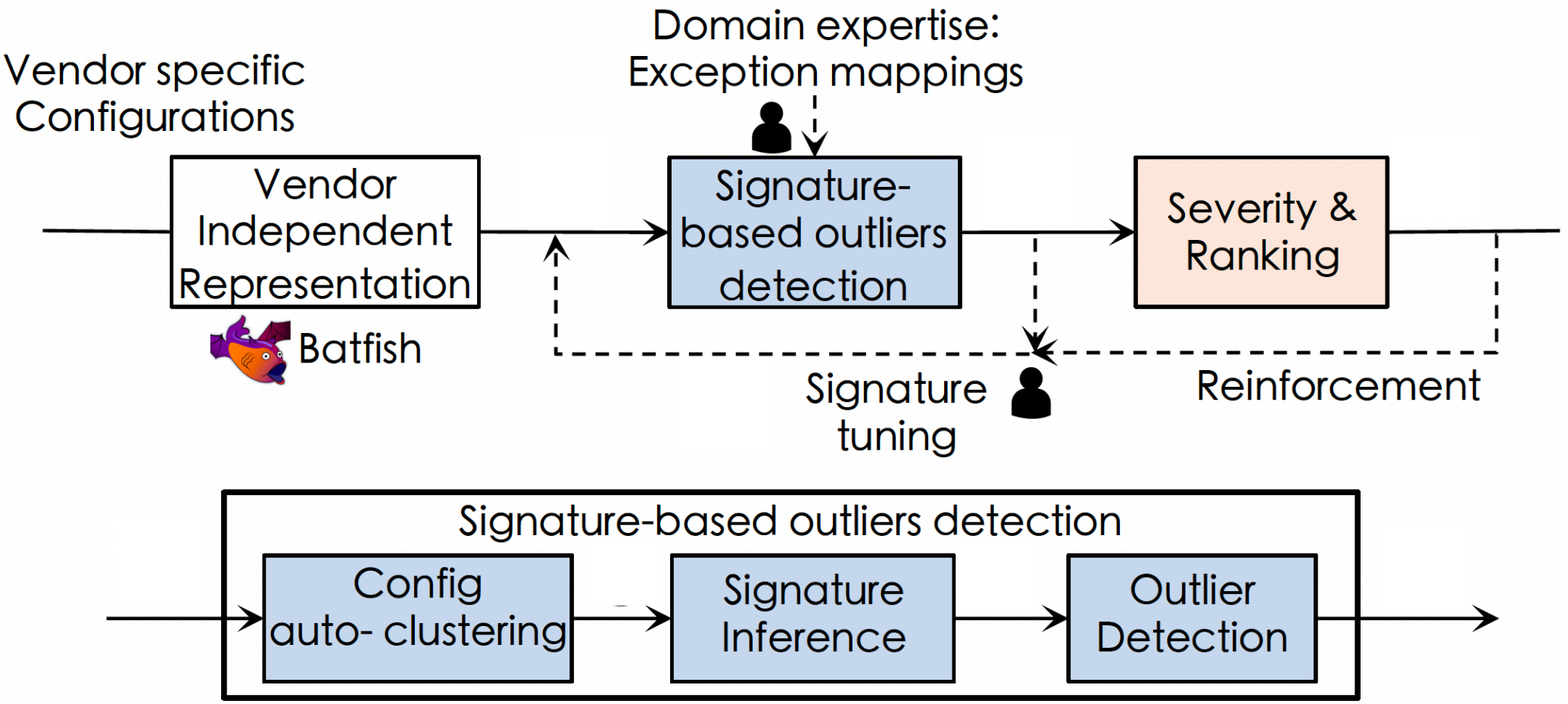
Key Steps Involved*:

- Auto Clustering for grouping configurations (ML + Statistical framework)
- Generate signatures/models out of each Cluster (Signature Inference + Domain expertise)
- Allow the configurations to be checked with respective signatures (ML Models + Signature retuning)
- Send outliers to the severity and ranking module for reprioritizing it (Domain expertise + ranking algorithms)

* For Brevity details not discussed, please refer to the paper and the Git repo for access to code + documentation

https://github.com/vasu018/outlier-analyzers

MAVERICK: High level architecture



MAVERICK: Ranked bug outcome and severity

Outlier	Signature Definition	Conformer Nodes	Outlier Definition	Outlier Nodes	Outlier Properties	Outlierness Value	Severity Score
outlier:Route_ Filter_List_0	{'action': [['PERMIT', 16]], 'ipWildcard':[['100.100.100.0/23', '*', 9], ... ['25-25', '*', 10]]}	['rt1-dc1', 'rt2-dc1'. .., rt91-dc1]	{'action': 'PERMIT', 'ipWildcard': '100.100.0.0/16', 'lengthRange': '16-20'}	['rt19-dc1', 'rt28-dc1']	[['lengthRange', '16-20']]	0.978	1.177

Columns (Description):

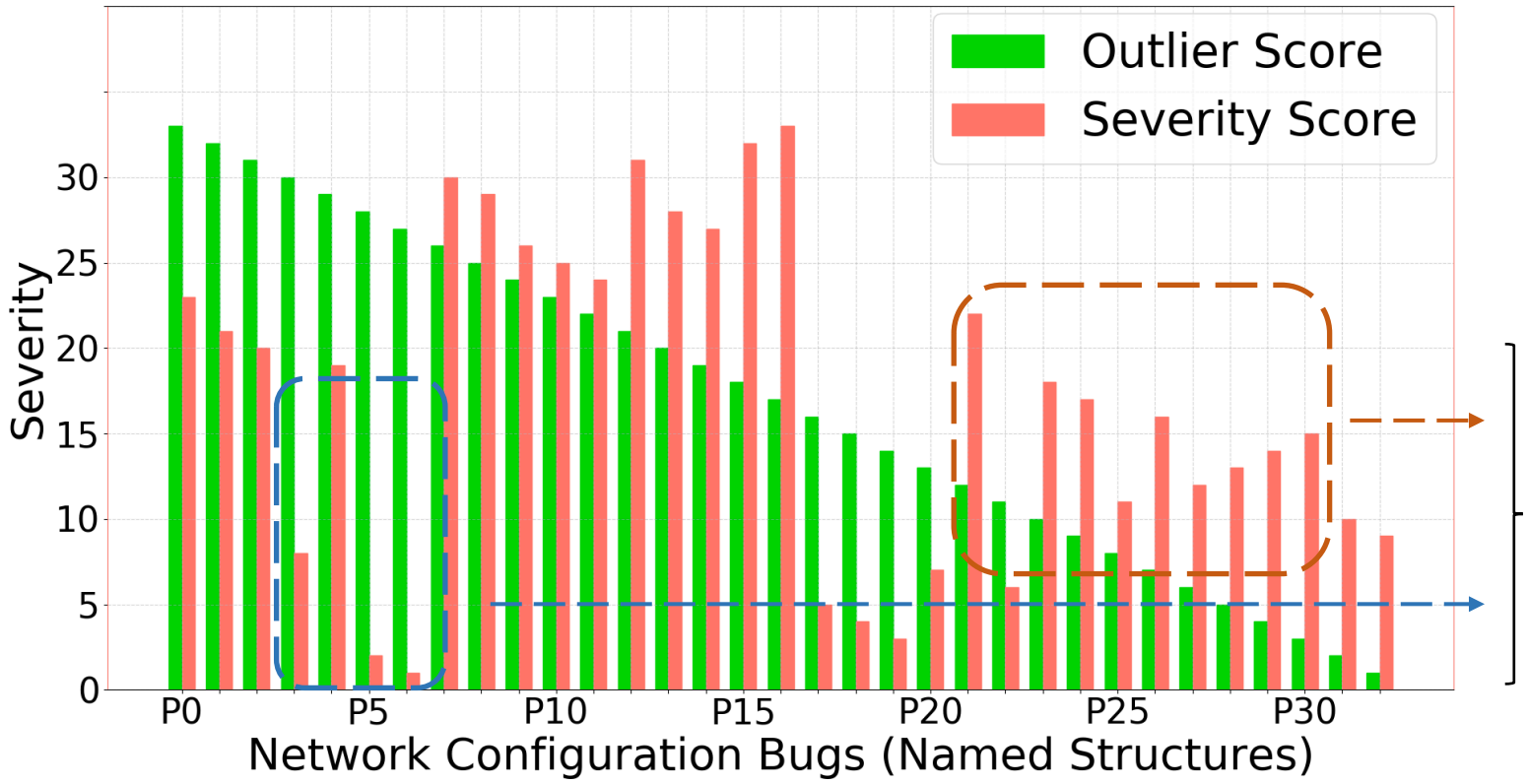
- **Outlier:** Describes the name of the outlier
- **Signature Definition:** Signature that is automatically inferred for bug detection with Maverick
- **Conformer Nodes:** Nodes that are inline with the Signature definition
- **Outlier Nodes:** Nodes that are deviant from the Signature definition
- **Outlier Definition:** Overall property that is deviant from the signature
- **Outlier Properties:** Specific property that is deviant from the actual signature
- **Outlierness Value:** Amount of deviation that defines the Outlier
- **Severity Score:** The score that defines the relative severity of the outlier compared to others

MAVERICK: Efficacy

Approach	TP	FP	FN	Precision	Recall
Z-score	392	1031	240	0.275	0.620
Modified Z-score	417	692	132	0.386	0.760
GMM	298	608	220	0.329	0.575
Maverick (Outliers)	472	74	32	0.864	0.937
Maverick (Retuning)	498	32	8	0.92	0.984

Higher precision and recall with simple signature tuning by administrator

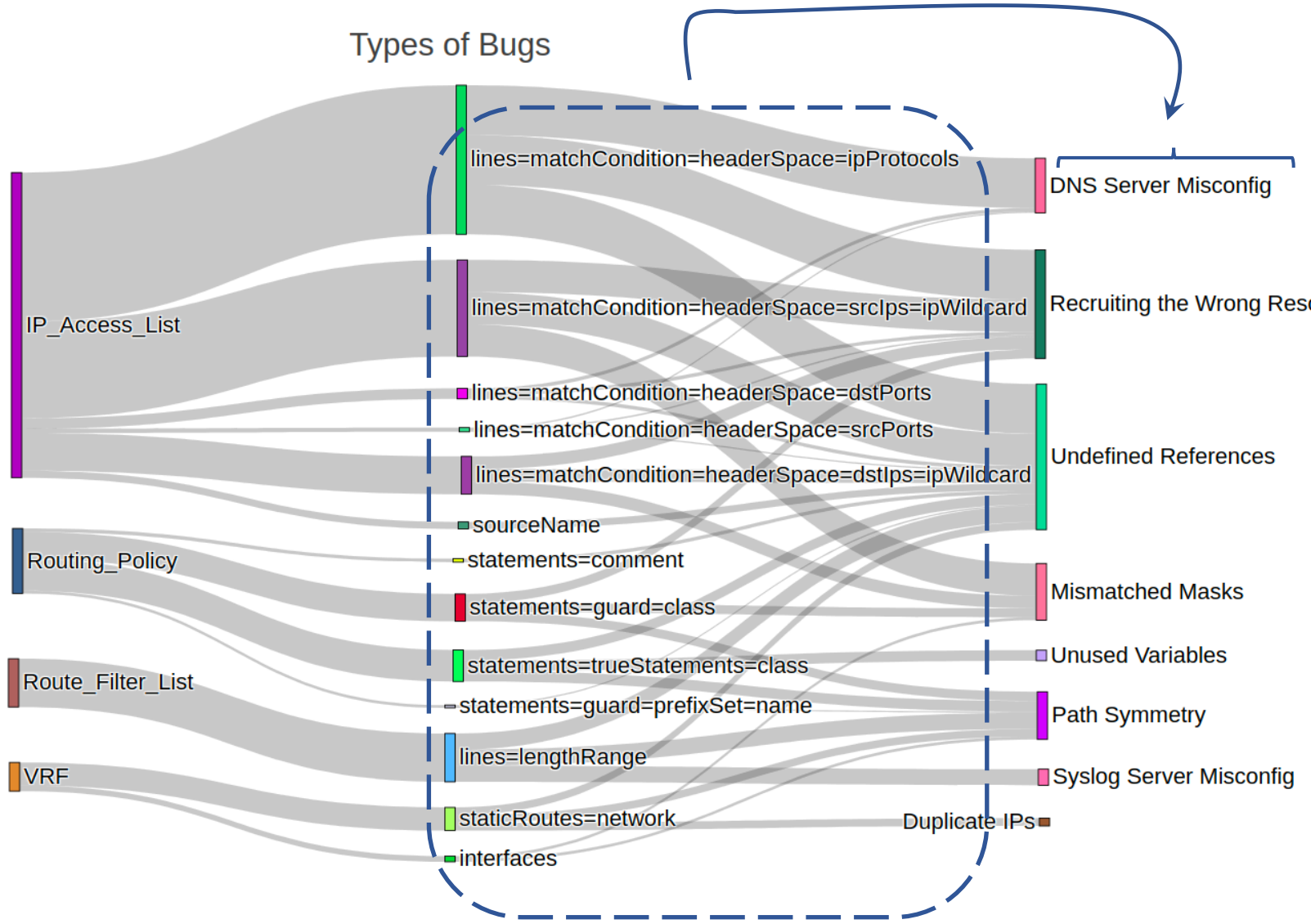
MAVERICK: Bugs discovered vs severity



Low outlier score ones are not required to low severe bugs.

Similarly, bugs with high outlier scores doesn't mean high severe bug.

MAVERICK: Outliers correlated to different bug types



Outliers mapped to different types of real-world bug types

MAVERICK

Git:
https://github.com/vasu018/outlier-analyzers

Contact: Vasudevan Nagendra
vnagendra@plume.com, vnagendra@cs.stonybrook.edu

Contributors:

1. Vasudevan Nagendra: https://www.linkedin.com/in/vasudevan-nagendra/
2. Abhishek Pokala: https://www.linkedin.com/in/abhishek-pokala/
3. Astitv Nagpal: https://www.linkedin.com/in/astitv-nagpal/
4. Omik Mahajan: https://www.linkedin.com/in/omik-mahajan/
5. Alfred Wu: https://www.linkedin.com/in/alfredwu2/
6. Sankalp Taralekar: https://www.linkedin.com/in/sankalp-taralekar/

Questions ?

Vasudevan Nagendra

vnagendra@plume.com,

vnagendra@cs.stonybrook.edu