Intrusion Detection in Spire

Maher Khan

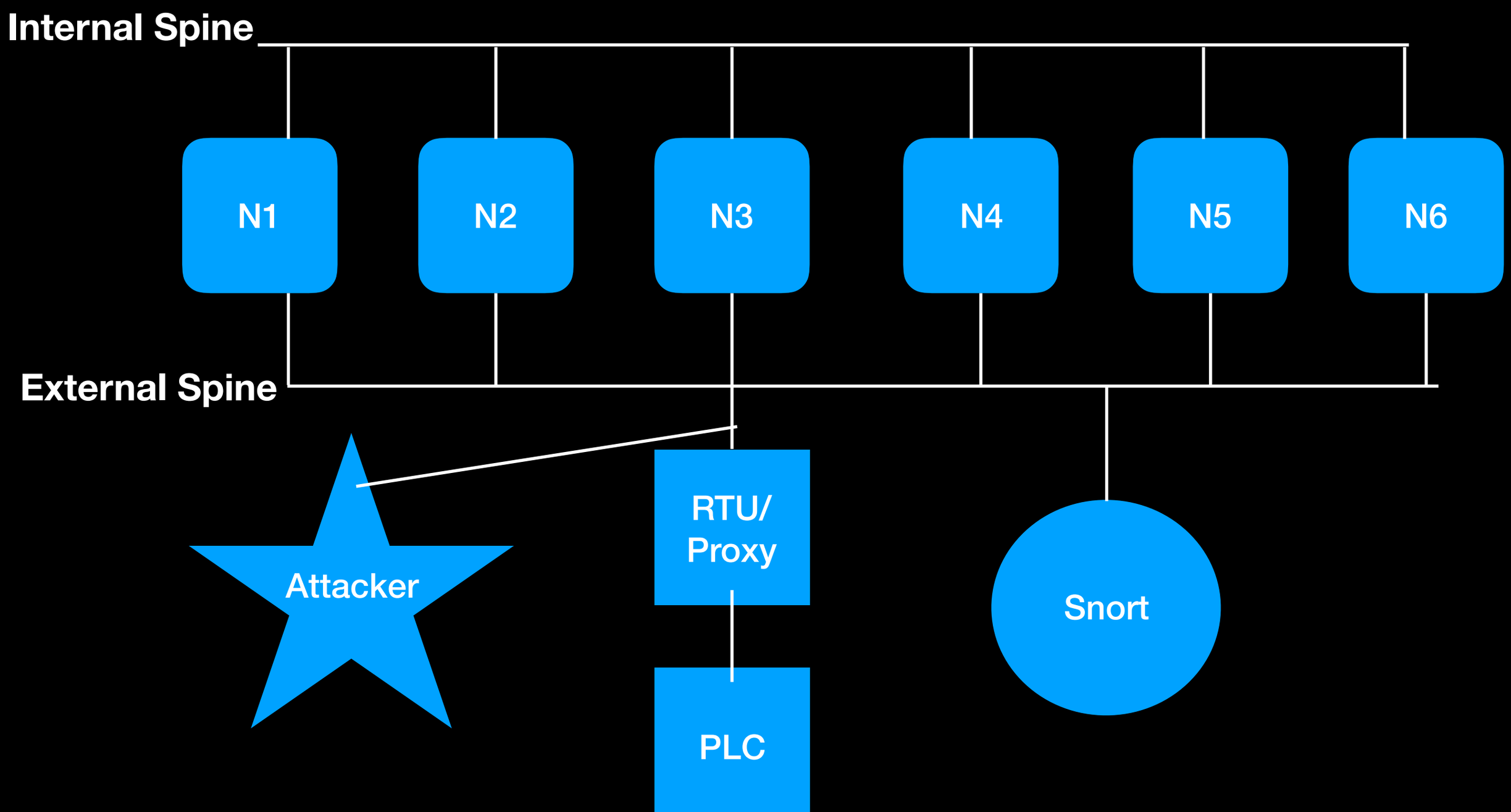
Problem

- Detect malicious messages and malicious senders in Spire (replicas or PLC proxies or PLCs)
- Improve intrusion tolerance through detection
- Explore and open up a line of research in combining intrusion detection with intrusion tolerance for Spire and other SCADA systems

Approach

- Researched on existing Intrusion Detection Systems
- Familiarize with Spire in terms of implementation and testing
- Setup, compile and run Spire on 6 VMs
- Integrated Snort (IDS) in the Spine network to analyze traffic
- Performed testing: DOS, ARP Poisoning and Replay

Design


Internal Spine
N1
N2
N3
N4
N5
N6
External Spine
Attacker
RTU/
Proxy
PLC
Snort

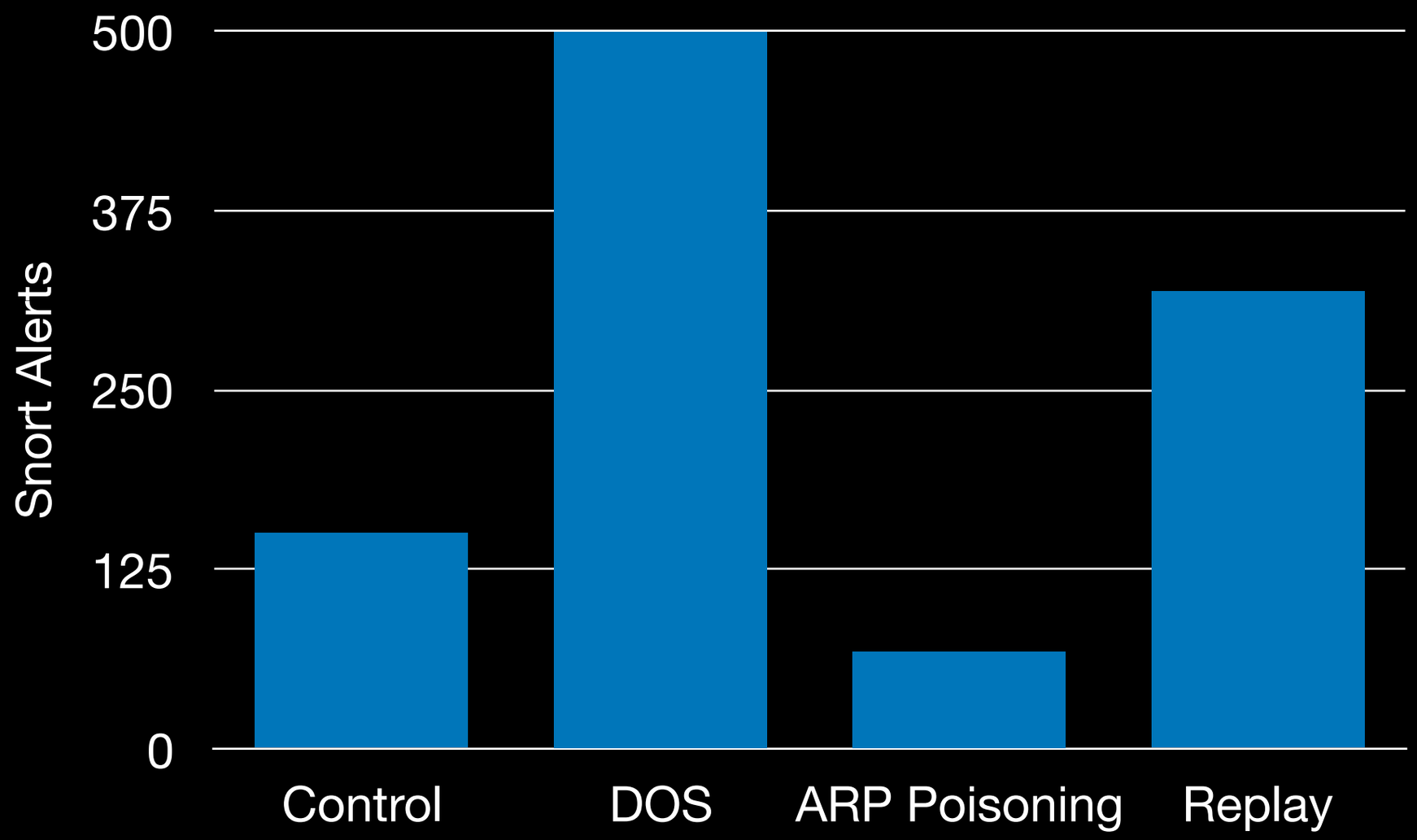
Implementation

- Each replica runs on a separate VM (6 in total)
- All the VMs connect with each other in a virtual network
- An additional VM to run Spire benchmark
- An additional VM to run Snort in promiscuous mode
- An additional VM to run attacks
- Collected stats from Snort will be monitored by administrator

Testing Model

- DOS attack
    - Flooded the RTU/Proxy with messages
- ARP Poisoning
    - Sending ARP messages so that traffic to and from RTU/Proxy is sent to the attacker instead
- Replay
    - Collect packets sent by the RTU/Proxy and replay them in the network

Results


500
375
250
125
0
Snort Alerts
Control
DOS
ARP Poisoning
Replay

Results

- The default snort rules produce a number of alerts in the normal scenario
- Currently the change in number of alerts tell us more about the change in traffic volume than actual attacks
- Since normal case traffic is regular, this can still indicate a problem
- We need to craft a set of rules in order for the normal case scenario to not throw any alert
- Snort should only throw alert when there an attack is likely, then this information can be used to detect attacks

How can we detect the attacks?

- DOS attacks are easier to identify because you can easily isolate the source sending the large number of packets to overwhelm a target

- ARP Poisoning is harder to identify since the attacker can secretly stay as a MITM without disrupting services
    - We can detect it by cross checking MACs in packets with an accepted list

- Replay attacks are the hardest to identify since the attacker does not change anything in the packet and simply reintroduces them in the network
    - This can be prevented by using session keys that expires after certain amount of time

Next Steps

- Modify the default Snort rules and come up with our own set of rules so that alerts are only generated in abnormal running conditions
- Identify attacks likely to occur in Spire
- Come up with IDS rules which in combination with intrusion tolerance of Spire can defend against those attacks

Demo