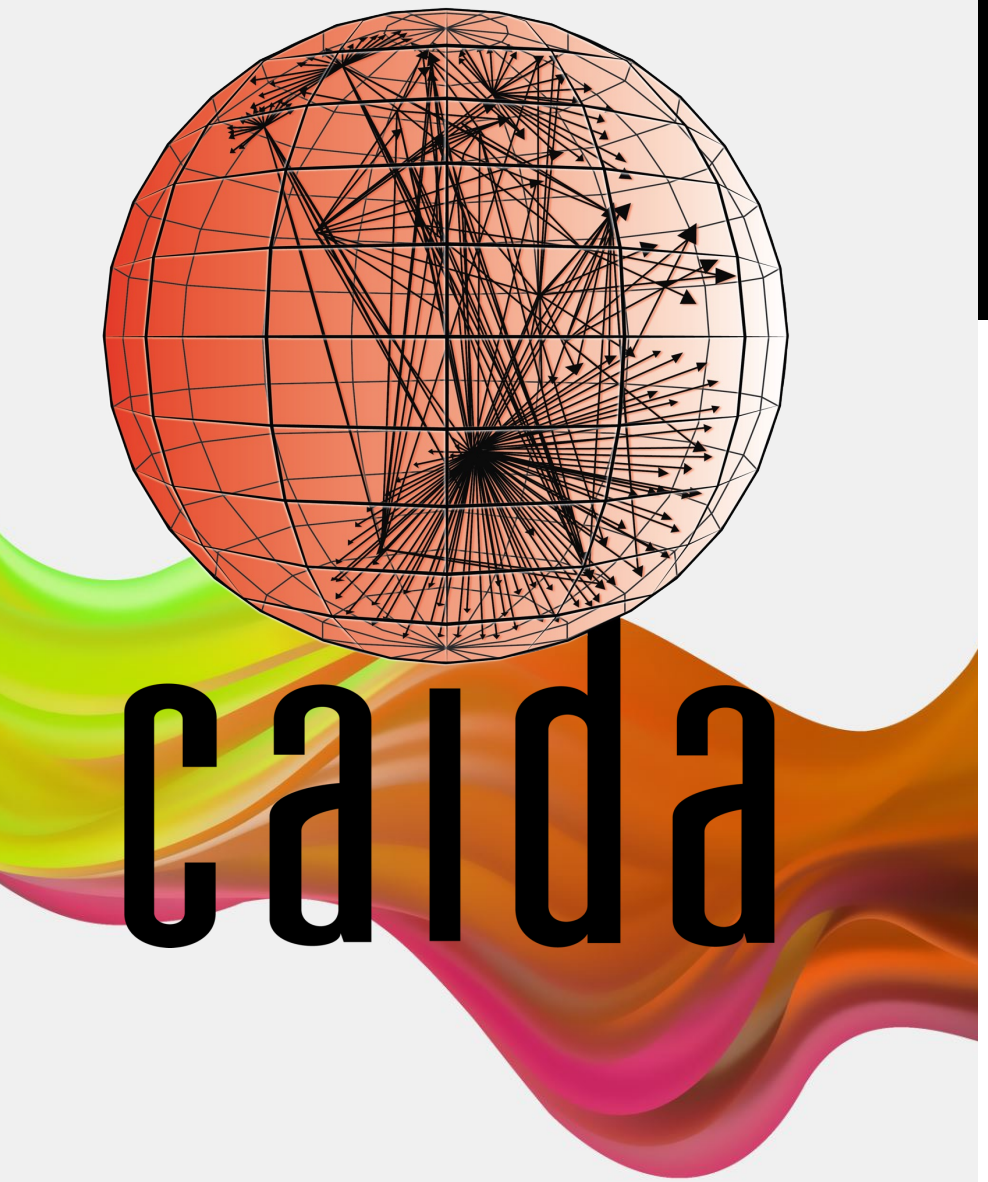


CSE 291
Internet data science for cybersecurity

1 March 2023

Traffic Data: Internet Background Radiation

Overview

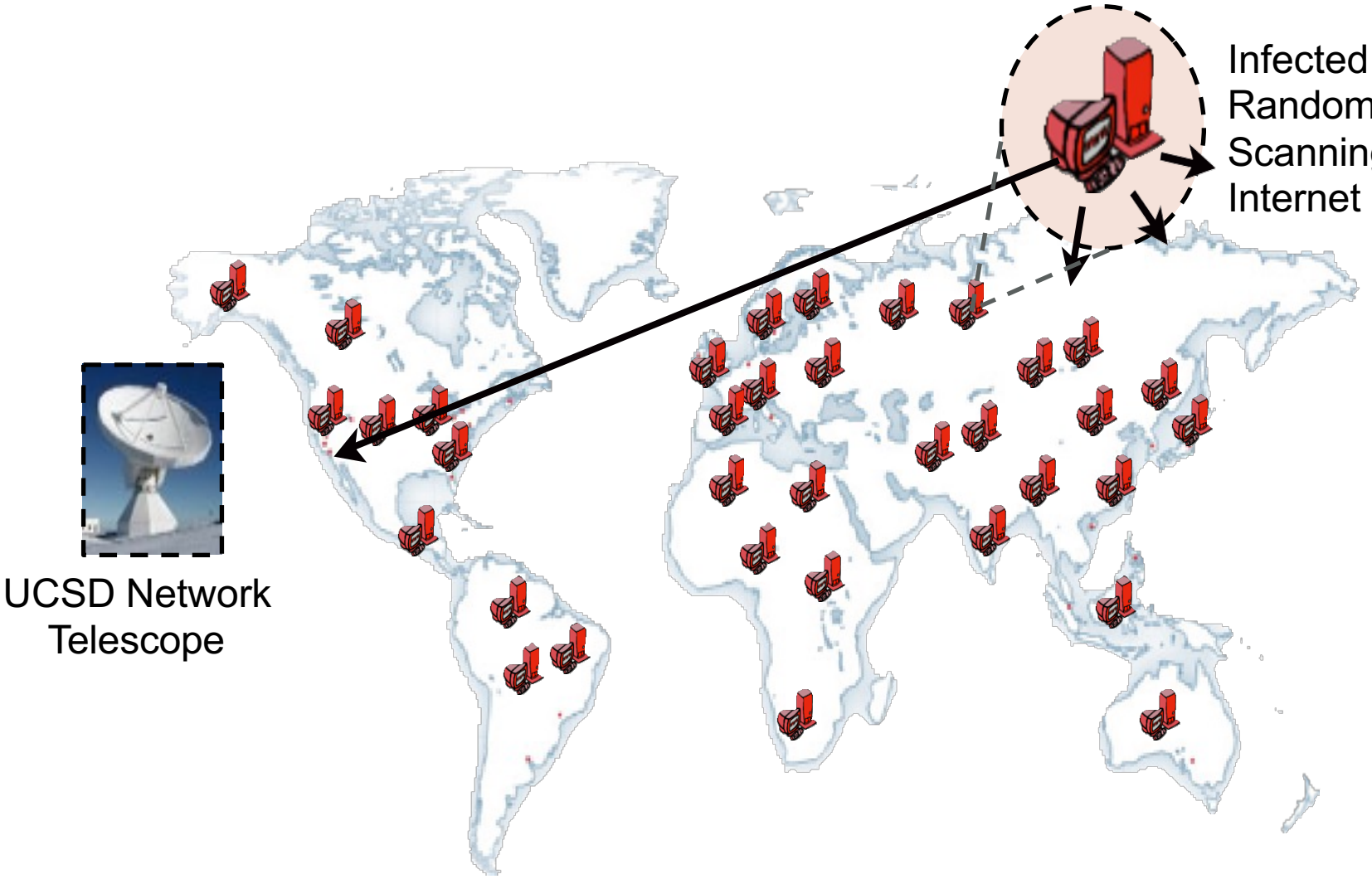
1. Background to support data science assignment in this class, without an assigned paper

Learning Objectives

1. Understand what is Internet background radiation
2. Understand how CAIDA's one-way passive traffic monitoring system works*
3. Analyze information contained in PCAP packets
4. Review assignment

*Derived from Alistair King's slides: "Internet Garbage, Storage, and Analysis"
https://www.caida.org/catalog/media/2014_internet_garbage_ndsaa/internet_garbage_ndsaa.pdf

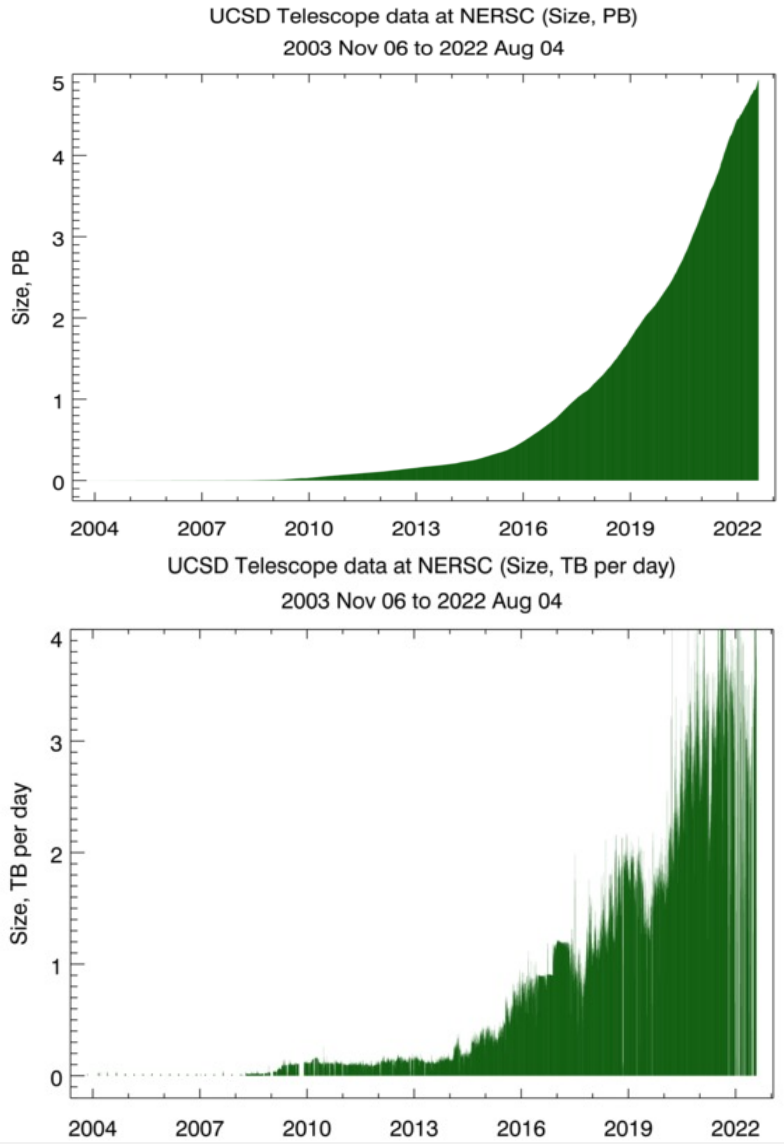
Internet Background Radiation (IBR)



Unsolicited traffic to unused ("dark") network:

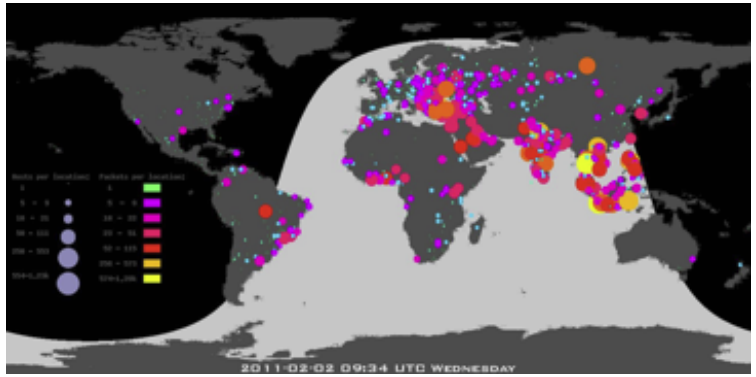
- Malware attempting to propagate
- Misconfiguration
- Network scans
- DDoS attacks
- "Backscatter"

UCSD Network Telescope (NT)

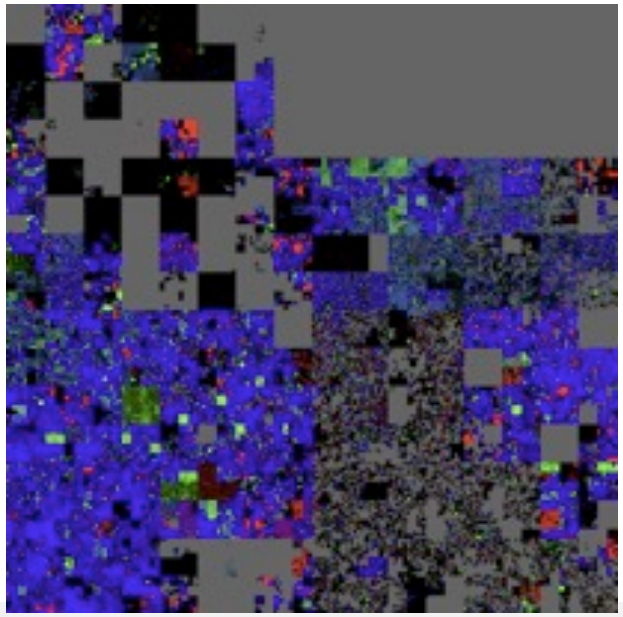


- Passive traffic monitoring system
- Instrumentation of darknet
- Globally routed, lightly utilized /9 and /10 (~1/350 of IPv4 address space)
- 24/7 full packet traces
- Archive of pcap data back to 2003
- Archived data volume ~ 5 PB
- Growth rate ~4 TB/day

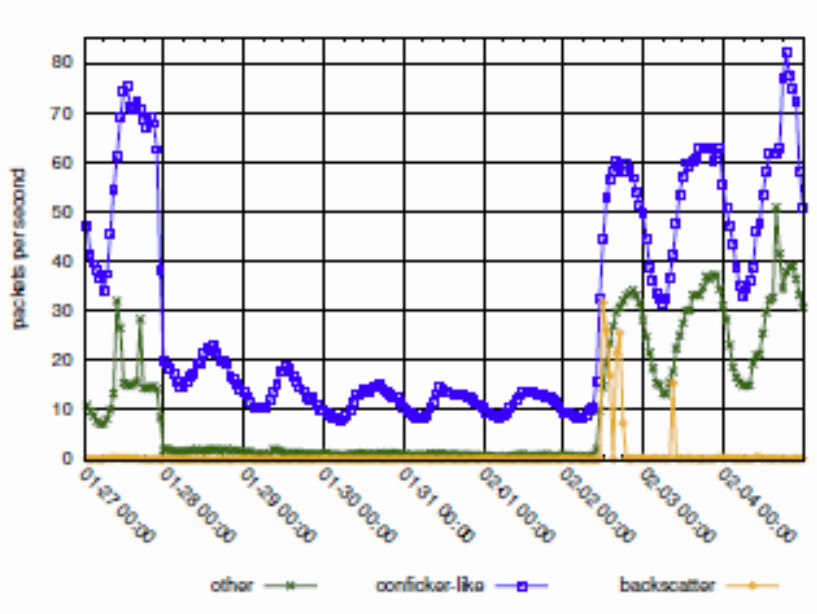
Telescope data use



Malware Spread

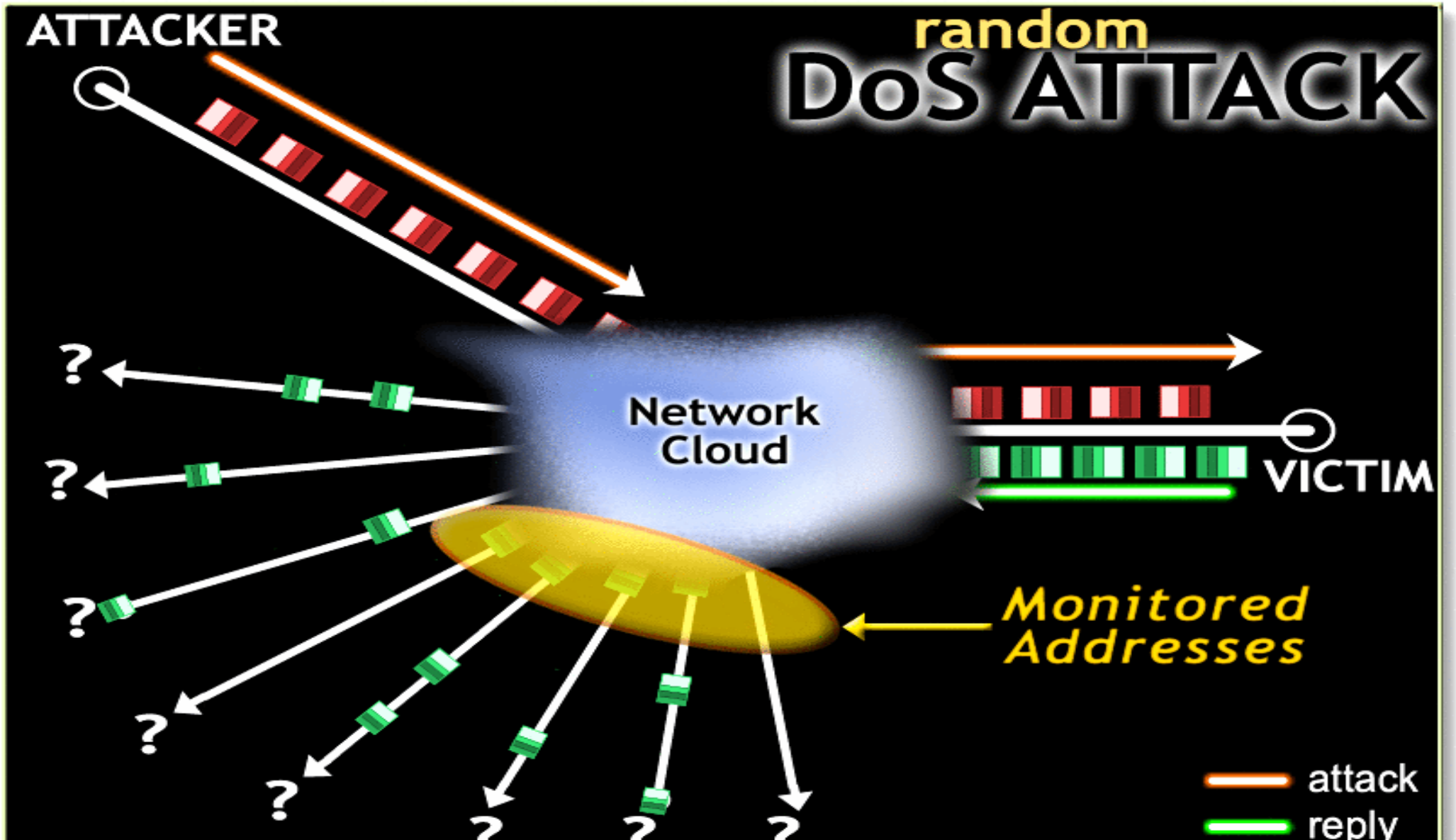


Testing ML models detecting/forecasting network anomalies



Connectivity Disruption

"Backscatter" (from DOS)



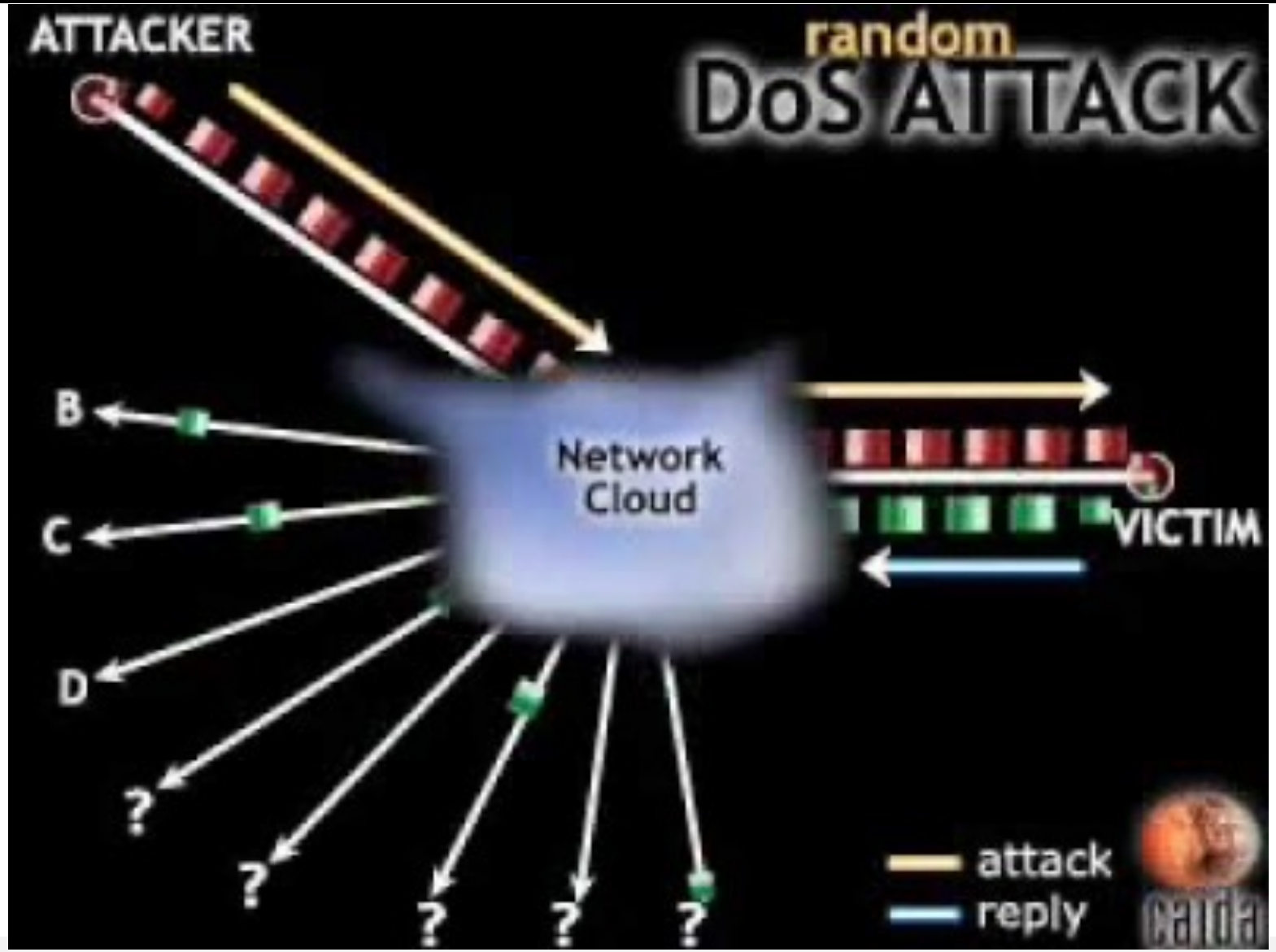
Denial of Service attack using *randomly spoofed* source IP addresses.

Triggers replies from victim address to random IPs

NT monitors many such IPs

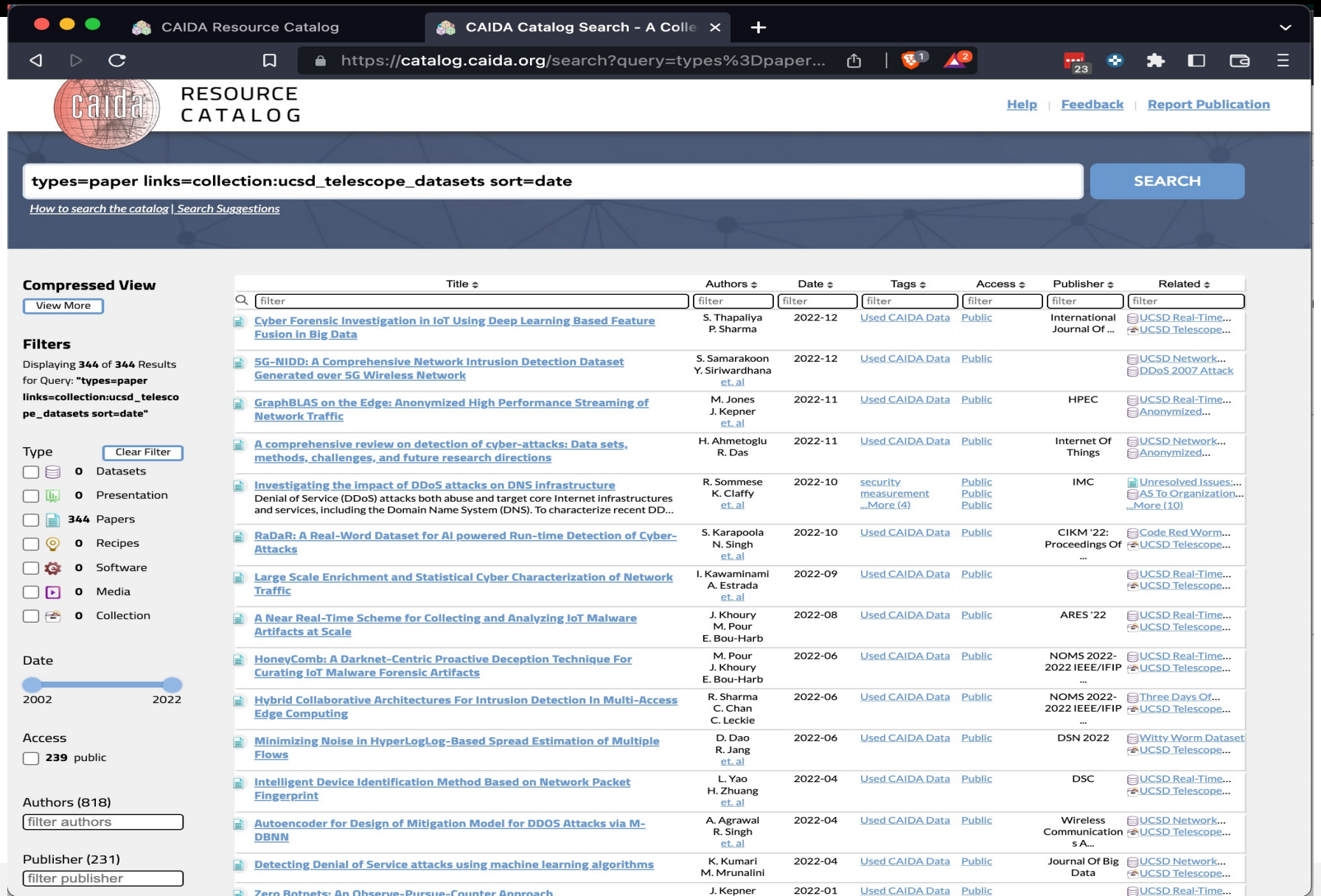
"Backscatter" (from DOS)

https://youtu.be/Yj_SdiG0RuY

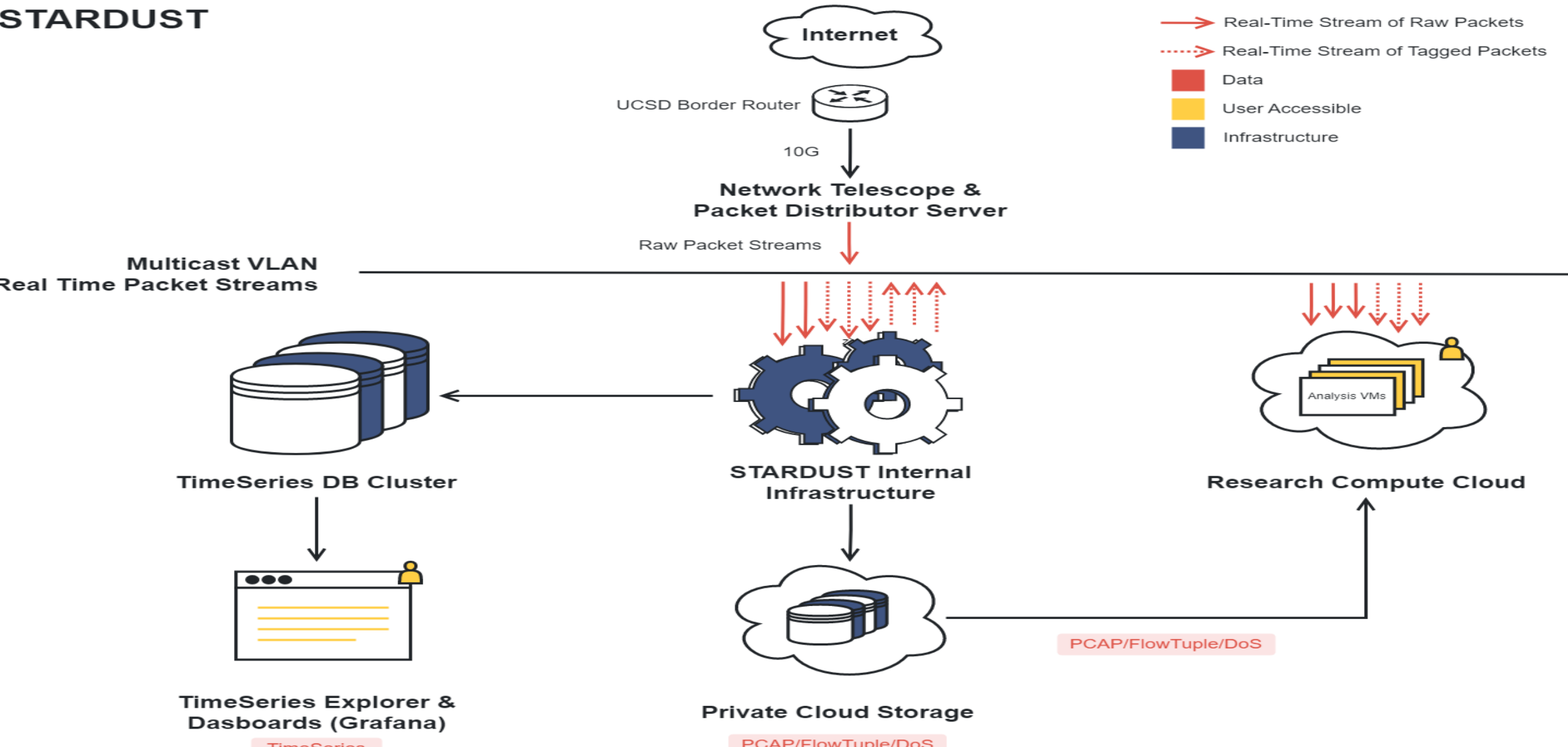


https://www.caida.org/projects/network_telescope/

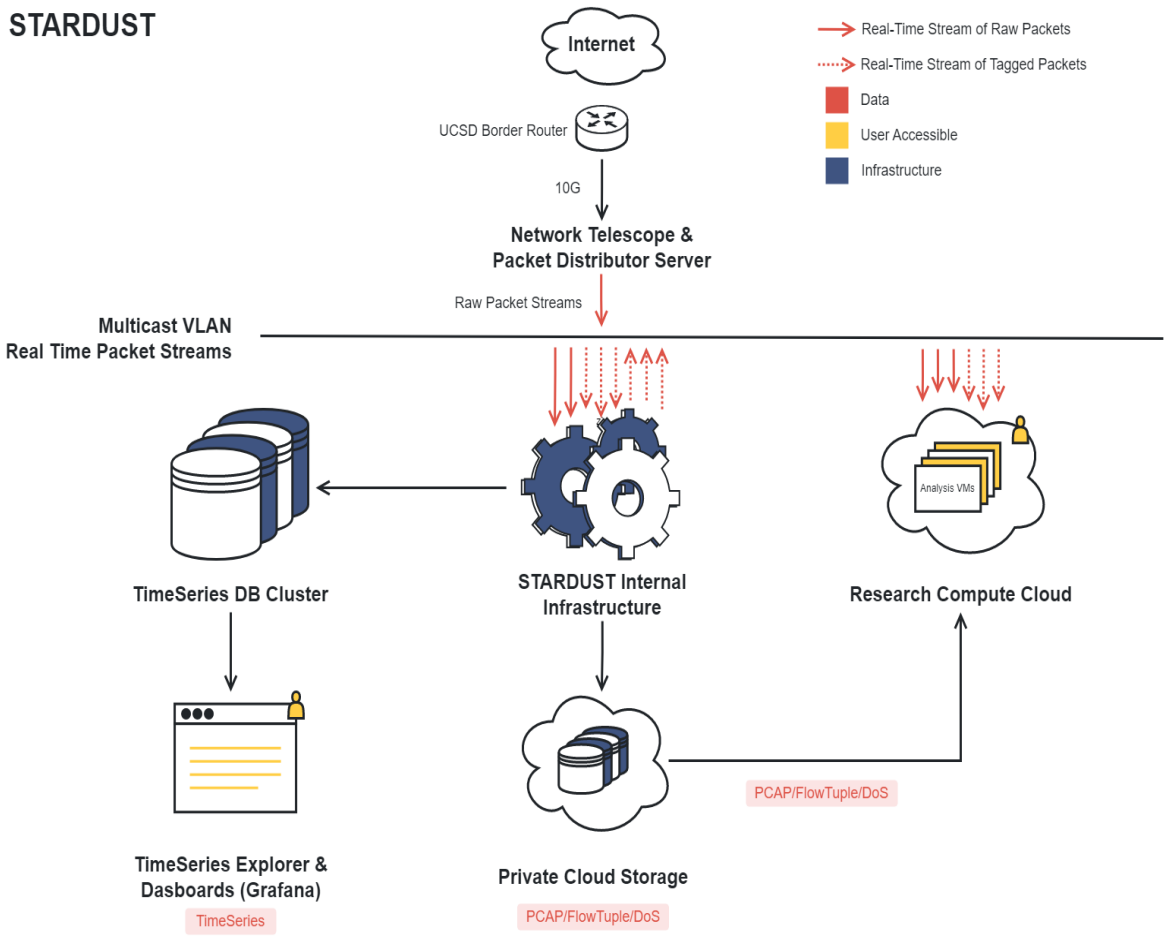
UCSD-NT Research (300+ papers)



Data Collection architecture

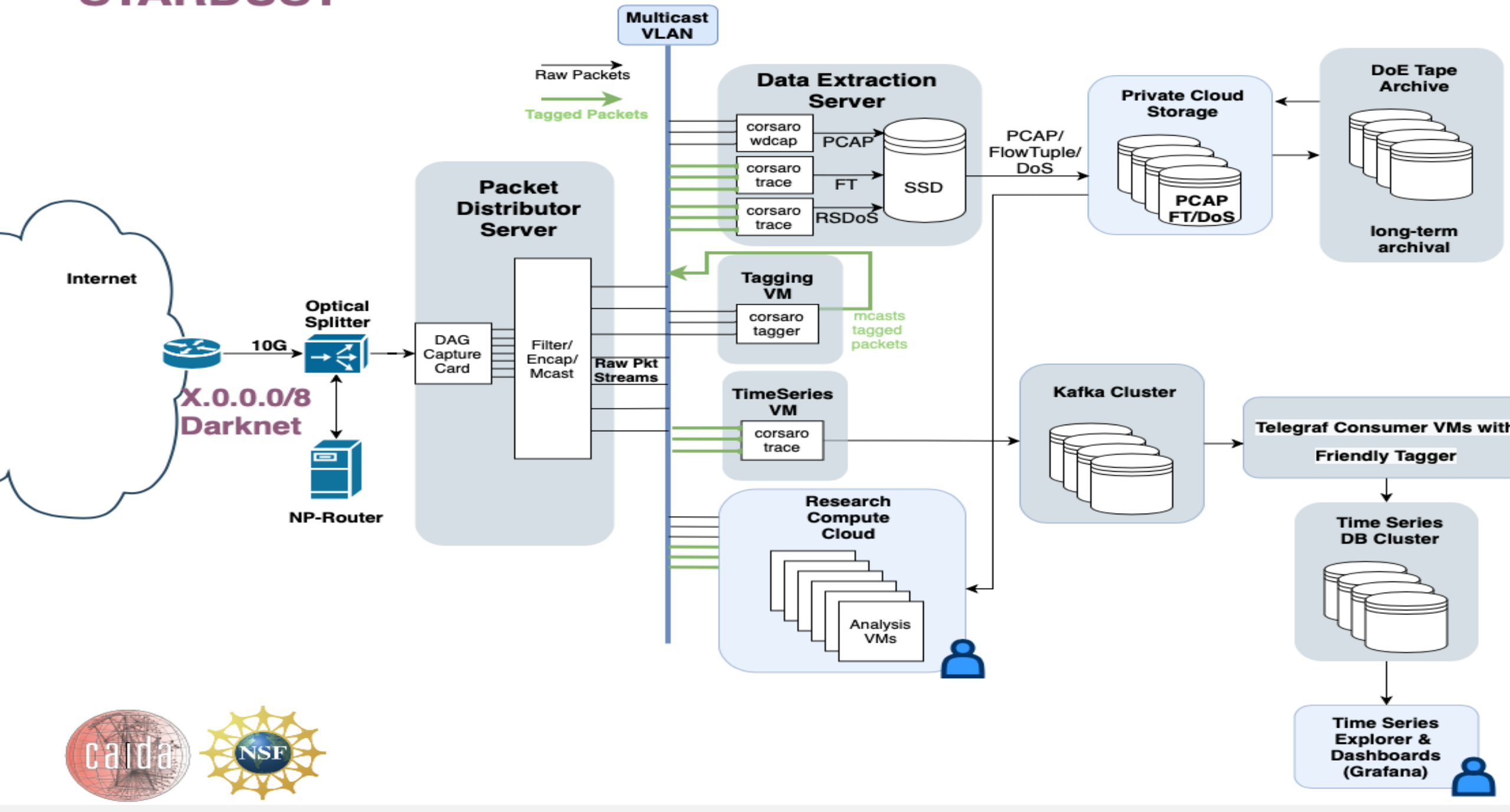


Data Collection architecture

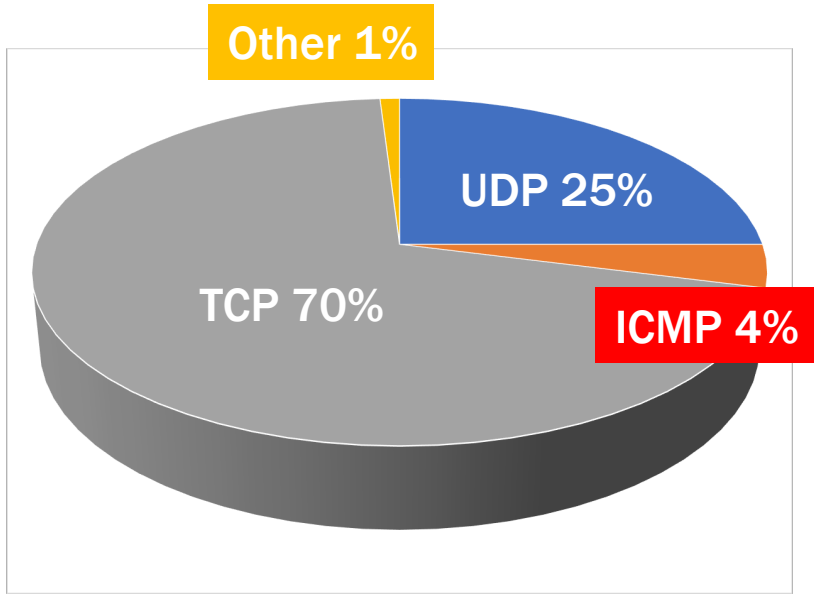


- Traffic captured by UCSD Network Telescope
- Sent to Packet Distribution Server
- nDAG Stream->dedicated VLAN, each via VMs
- Traffic processed and enriched with metadata (geolocation, source IP address ASN, …) , re-distributed on same VLAN, separate stream
- Raw and flow-level traces stored in cloud-base object storage, accessible via VMs
- Additional tools, APIs and software for other means of data access
- Traffic processed to extract statistics
- statistics: per-minute counts of unique source IPs/country, Ases, protocols, port, on time-series visual dashboard

STARDUST


Multicast VLAN
Raw Packets
Tagged Packets
Data Extraction Server
corsaro wdcap
PCAP
corsaro trace
FT
corsaro trace
RSDoS
SSD
PCAP/ FlowTuple/ DoS
Private Cloud Storage
PCAP FT/DoS
DoE Tape Archive
long-term archival
Packet Distributor Server
Internet
Optical Splitter
10G
X.0.0.0/8 Darknet
NP-Router
DAG Capture Card
Filter/ Encap/ Mcast
Raw Pkt Streams
Tagging VM
corsaro tagger
mcasts tagged packets
TimeSeries VM
corsaro trace
Kafka Cluster
Telegraf Consumer VMs with
Friendly Tagger
Time Series DB Cluster
Research Compute Cloud
Analysis VMs
Time Series Explorer & Dashboards (Grafana)

Packet capture



- X.0.0.0/9 and X.0.0.0/10 routed over 10G link
- Packet velocity is a challenge

Hourly pcap files

- Collected Traffic is written to **pcap** trace files
- Unanonymized
- Not truncated (includes payload)
- pcap file contains 1 hour of data Size ~100 GB
- ~ 30-60 latest days stored locally in Swift object storage
- Older pcaps stored up at NERSC (DOE-funded archival)

Processing tools: Libtrace

https://github.com/LibtraceTeam/libtrace

https://stardust.caida.org/docs/tutorials/libtracetutorial/

Example of tracepkdump pcapfile output
tracepkdump pcapfile:location/name |head

```
Capture: Packet Length: 60/64 Direction Value: -1
Ethernet: Dest: 3c:fd:fe:19:d8:00 Source: 00:de:fb:ba:06:c7 Ethertype: 0x0800
IP: Header Len 20 Ver 4 DSCP 00 ECN 0 Total Length 40
IP: Id 54321 Fragoff 0
IP: TTL 241 Proto 6 (tcp) Checksum 46313
IP: Source 45.153.203.175 Destination 44.28.27.80
TCP: Source 43922 Dest 82
TCP: Seq 2846108233
```

Example of "print the first n packet in the file"
tracepkdump –c <number of packets> pcapfile:location/name

```
apture: Packet Length: 60/64 Direction Value: -1
thernet: Dest: 3c:fd:fe:19:d8:00 Source: 00:de:fb:ba:06:c7 Ethertype: 0x0800
P: Header Len 20 Ver 4 DSCP 00 ECN 0 Total Length 40
P: Id 65439 Fragoff 0
P: TTL 242 Proto 6 (tcp) Checksum 37144
IP: Source 45.129.33.49 Destination 44.111.188.246
TCP: Source 40961 Dest 3428
TCP: Seq 1117343245
TCP: Ack 0
TCP: DOFF 5 Flags: SYN Window 1024
TCP: Checksum 13759 Urgent 0
unknown protocol tcp/3428
Unknown Protocol: 3428
 00 00 e9 75 10 0a                                    ...u..
```

corsarotrace

https://github.com/CAIDA/corsaro3

https://stardust.caida.org/docs/tutorials/corsaro3/

- Packet processing tool within corsaro3 to run custom analysis code against captured packets
- Leverages *libtrace*
- Supports parallel processing using multiple threads
- Allows running custom analysis routines through a plugin-based system (in C)

Corsaro

- Easily extensible, plugin architecture
- Per-packet operations:
  - IP Geolocation
  - IP to AS mapping
  - CryptoPan Anonymization
  - Metadata-based packet filtering
- Per-interval operations:
  - Compute per-interval statistics
  - Write out aggregated data

Flowtuples

- Pcaps from the UCSD-NT are massive
  - Processing takes a long time, storage fills up quickly.

- Solution: Aggregate similar packets into "flowtuples"

  Retain key header fields: Src IP, Dest IP, Src Port, Dest Port, Protocol, TTL, TCP Flags, IP Len

  Supports many telescope use cases

  Per-minute

  Serialized in binary format

  Easier to use and share

- Regular flowtuples from UCSD-NT since 2008
- Formats changed over time: https://github.com/CAIDA/corsaro3/wiki/Flowtuple-Formats

Flowtuple v3

~2018: changed format to add useful meta-data to flowtuple records

- Use mainstream big data format (Avro)
- Create tools / APIs for simplifying large-scale flowtuple analysis

src_ip	dst_ip	src_port	dst_port	protocol	ttl	tcp_flags	ip_len	packet_cnt
1.2.3.4	10.100.0.45	44120	22	6	122	2 (SYN)	52	4

tcp_synlen	tcp synwinlen	is_spoofed	is_masscan	maxmind continent	maxmind country	netacq continent	netacq country	prefix2asn
32	65535	False	False	AS	KR	AS	KR	123456

Shane Alcock, "Flowtuples IV: Reality Strikes Back",
https://www.caida.org/catalog/media/2021_flowtuples_iv_dust/flowtuples_iv_dust.pdf

Flowtuple v3

Storage requirements Unsustainable

- 160 TB per year, without factoring in traffic growth over time
  - 300 MB per minute in 2020
  - 40 MB per minute in 2015
  - 1 MB per minute in 2008

Shane Alcock, "Flowtuples IV: Reality Strikes Back",
https://www.caida.org/catalog/media/2021_flowtuples_iv_dust/flowtuples_iv_dust.pdf

Flowtuple v3 ⟹ v4

Specific destination IPs in dark space not so important
- Are packets hitting few addresses (or subnets) or many?
- Consider a scan of an entire /16 subnet by a single host
  - In FT3, this will result in 65536 near-identical flow records
  - Need 1 record that includes # addresses scanned

Source ports for unsolicited traffic are generally ephemeral
- (In)consistency of (source) port of interest, not specific values

Apply same logic to TTLs, packet sizes and TCP flags
- Is value consistent? If so, what are those values?

Shane Alcock, "Flowtuples IV: Reality Strikes Back",
https://www.caida.org/catalog/media/2021_flowtuples_iv_dust/flowtuples_iv_dust.pdf

Flowtuple v4

Aggregate to 5-minute (not 1-minute) granularity, and…

- Destination subnets (/16s) instead of destination IP addresses
- Source port, TTL, packet size and TCP flags no longer part of flow key
- Instead, record “common” values, # unique values seen for the flow
- Update processing tools

src_ip	dst_net	dst_port	protocol	packet_cnt	uniq_pkt_sizes	uniq_ttls	uniq_dst_ips
1.2.3.4	10.100.0.0	22	6	518	7	1	232

uniq_src ports	uniq_tcp flags	first_syn length	first_tcp rwin	maxmind continent	maxmind country	netacq continent	netacq country	prefix2asn
475	1	32	65535	AS	KR	AS	KR	123456

common pktsizes	common pktsize_freqs	common ttls	common ttl_freqs	common srcports	common srcport_freqs	common tcpflags	common tcpflag_freqs
[72, 80]	[356, 141]	[122]	[518]	[ ]	[ ]	[2]	[518]

Shane Alcock, "Flowtuples IV: Reality Strikes Back",
https://www.caida.org/catalog/media/2021_flowtuples_iv_dust/flowtuples_iv_dust.pdf

For further details

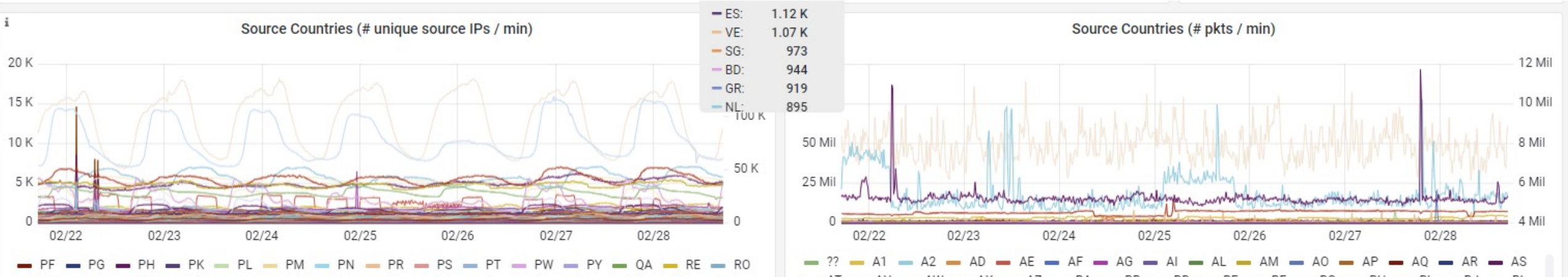
https://www.caida.org/catalog/media/2021_flowtuples_iv_dust/flowtuples_iv_dust.pdf

Starting slide 17

Time Series data

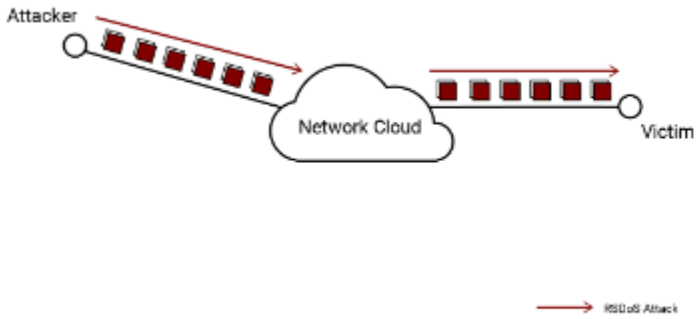
https://explore.stardust.caida.org/

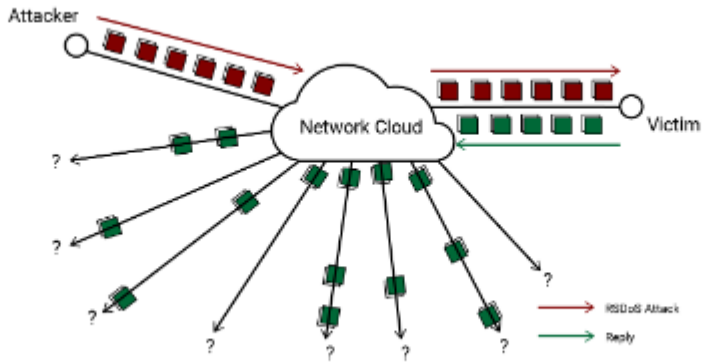
- Illustrates change of one variable over time
- Variables include:
  - packets per second
  - bits per second
  - unique source IPs per minute
  - unique source ASNs per minute
  - unique destination IPs per minute
- (Live demo!)

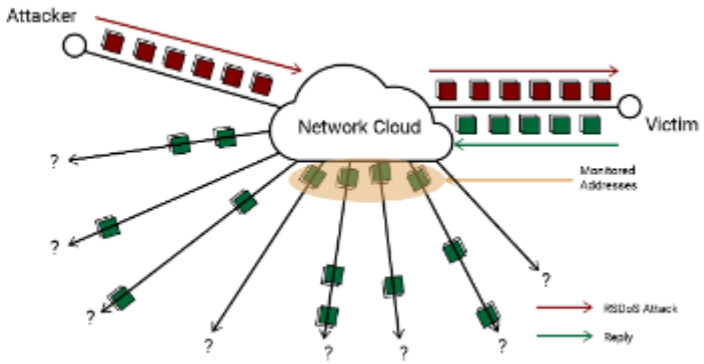


Denial-of-service attacks

- **NT Telescope data helps to detect** and observe the randomly-spoofed distributed denial-of-service (**RSDoS**) attacks
- Fake source IP may be used
- DoS attack victim can't distinguish between legit and attacker's requests -- > responds to every received request
- If attacker spoofs source address in NT Telescope it sends respond to unused IPs in this network → we detect attacks.







RSDoS data

- Meta-data of RSDoS from the backscatter packets collected by the UCSD NT Telescope.
- Aggregated from the raw telescope data
- Updated daily
- Generated by processing 5-minute intervals of raw data and extracting the response packets sent by victims of RSDoS attacks
- Corsaro3 → Apache Avro format
- Each record describes one attack observed within 5-minute interval

RSDoS record format

Property	Data Type	Description
bin_timestamp	long	The timestamp for the interval that this attack was observed in.
initial_packet_len	int	The size of the first packet observed as part of this attack.
target_ip	long	The IP address of the address that was the target of the DoS attack (i.e. the source address of the observed packets). Encoded as a 32 bit integer.
target_protocol	int	The transport protocol used for the attack (1 = ICMP, 6 = TCP, 17 = UDP).
attacker_slash16_cnt	long	The number of distinct /16 subnets in our monitored network that received packets from the victim.
attack_port_cnt	long	The number of unique source ports used by the attacker (i.e. the number of unique destination ports seen on received packets attributed to this attack).
target_port_cnt	long	The number of unique ports that were targeted on the victim (i.e. the number of unique source ports seen on received packets attributed to this attack).
packet_cnt	long	The number of packets that were attributed to this attack.
icmp_mismatches	long	The number of ICMP packets attributed to this attack where the source IP address in the body of the ICMP packet (e.g. the original datagram reflected in a Destination Unreachable message) does not match the IP address that the ICMP packet was sent to.
byte_cnt	long	The number of bytes that have been sent to our network due to this attack (based on IP length).
max_ppm_interval	long	The peak observed packet rate observed for this attack.
start_time_sec	long	The seconds portion of the Unix timestamp of the first packet attributed to this attack.
start_time_usec	int	The microseconds portion of the Unix timestamp of the first packet attributed to this attack.
latest_time_sec	long	The seconds portion of the Unix timestamp of the last packet attributed to this attack.
latest_time_usec	int	The microseconds portion of the Unix timestamp of the last packet attributed to this attack.
first_attack_port	int	The source port that was used by the first packet that was attributed to this attack.
first_target_port	int	The destination port that was used by the first packet that was attributed to this attack.
maxmind_continent	string	The continent where the target IP address is located, according to Maxmind geo-location data.
maxmind_country	string	The country where the target IP address is located, according to