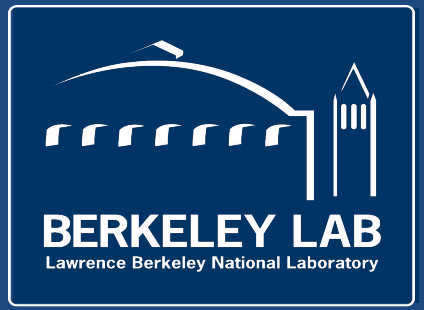
Operationalizing Zeek Deployments

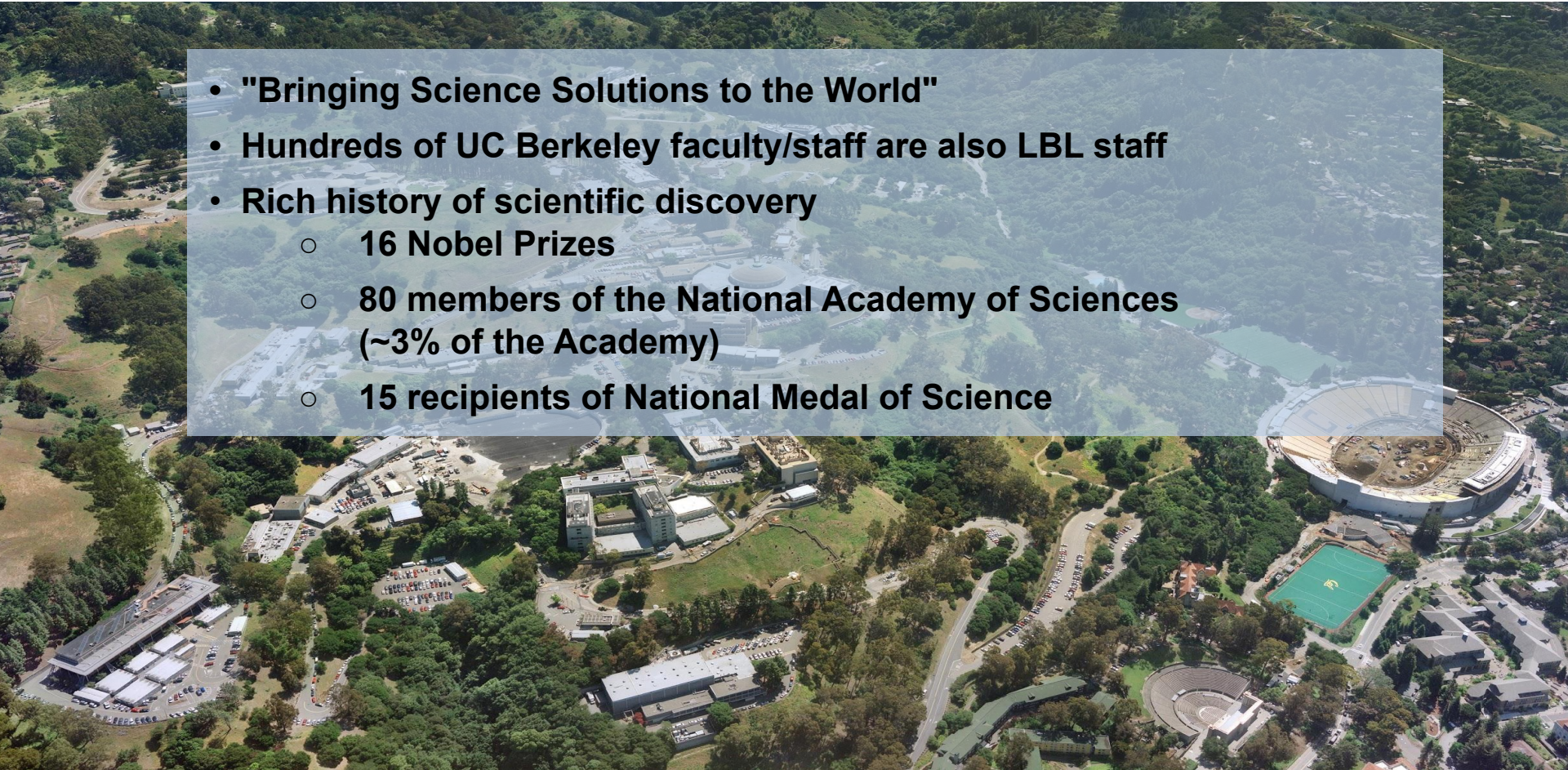
James Welcher, Aashish Sharma
Lawrence Berkeley National Lab

U.S. DEPARTMENT OF
ENERGY

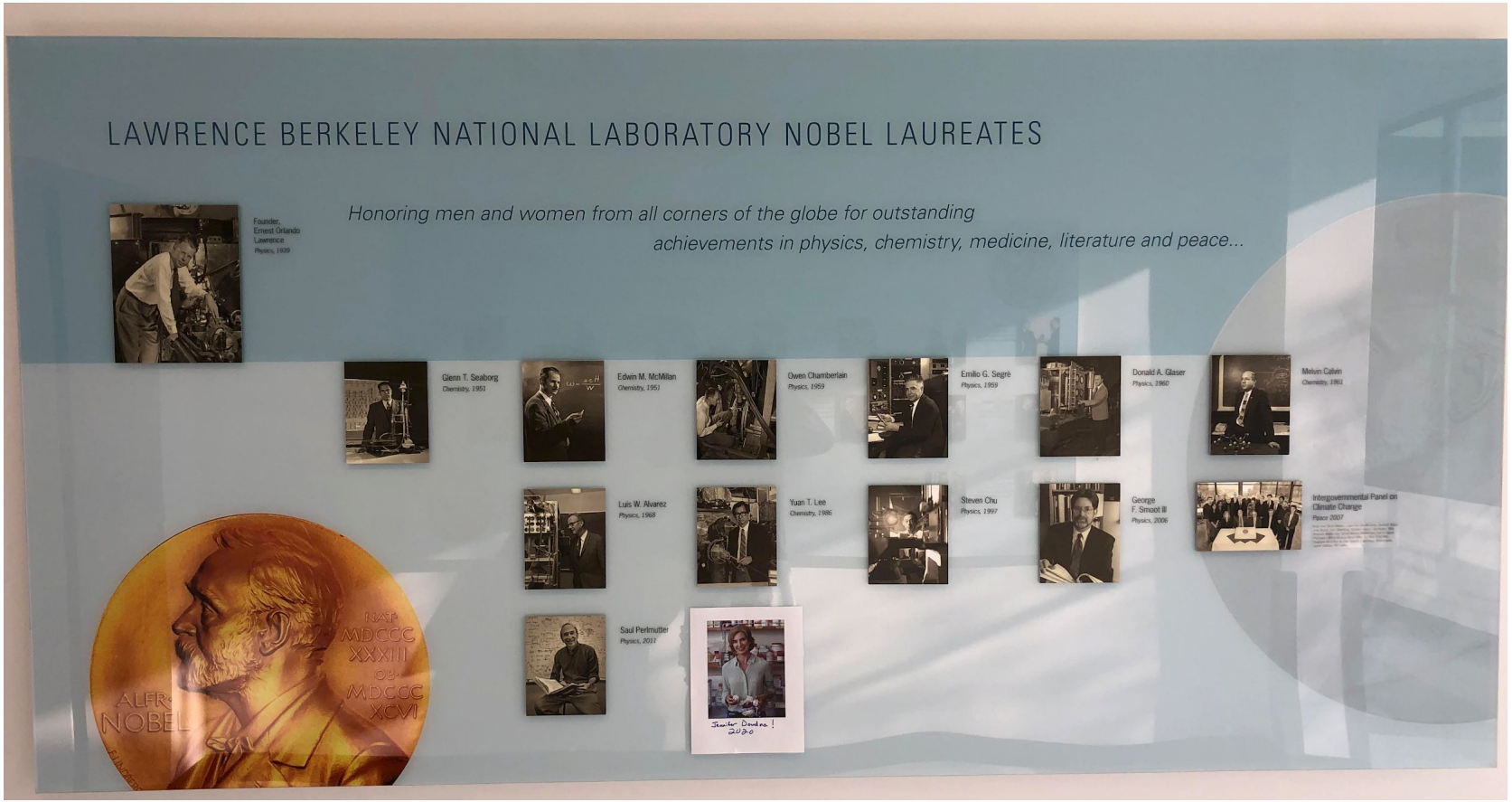
UNIVERSITY OF
CALIFORNIA

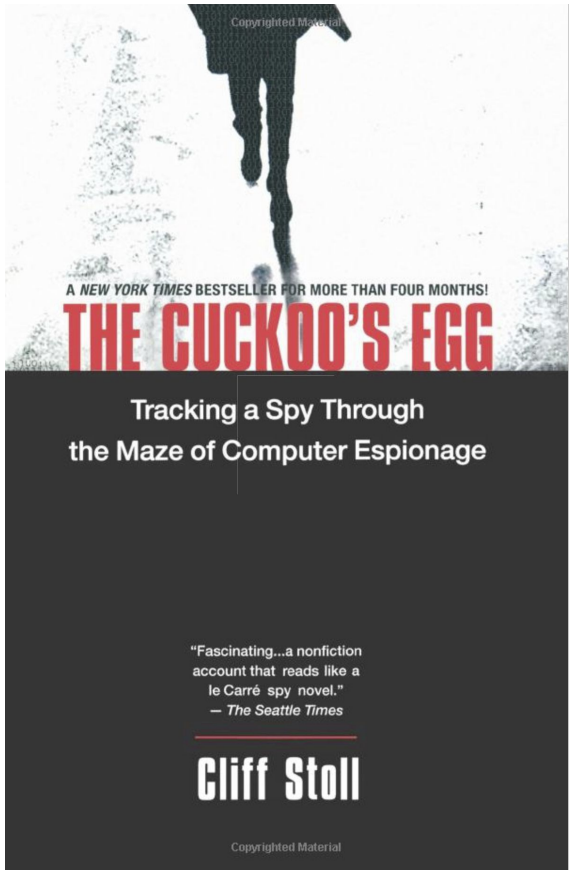


Lawrence Berkeley National Laboratory



- "Bringing Science Solutions to the World"
- Hundreds of UC Berkeley faculty/staff are also LBL staff
- Rich history of scientific discovery
  - 16 Nobel Prizes
  - 80 members of the National Academy of Sciences (~3% of the Academy)
  - 15 recipients of National Medal of Science


LAWRENCE BERKELEY NATIONAL LABORATORY NOBEL LAUREATES
Founder, Ernest Orlando Lawrence
Physics, 1939
Honoring men and women from all corners of the globe for outstanding achievements in physics, chemistry, medicine, literature and peace...
Glenn T. Seaborg
Chemistry, 1951
Edwin M. McMillan
Chemistry, 1951
Owen Chamberlain
Physics, 1959
Emilio G. Segrè
Physics, 1959
Donald A. Glaser
Physics, 1960
Melvin Calvin
Chemistry, 1961
Luis W. Alvarez
Physics, 1968
Yuan T. Lee
Chemistry, 1986
Steven Chu
Physics, 1997
George F. Smoot III
Physics, 2006
Intergovernmental Panel on Climate Change
Peace 2007
Saul Perlmutter
Physics, 2011
NATP MDCCC XXXIII OB MDCCC XCVI
ALFR NOBEL



Network utilities from Site

- traceroute
- libpcap
- tcpdump

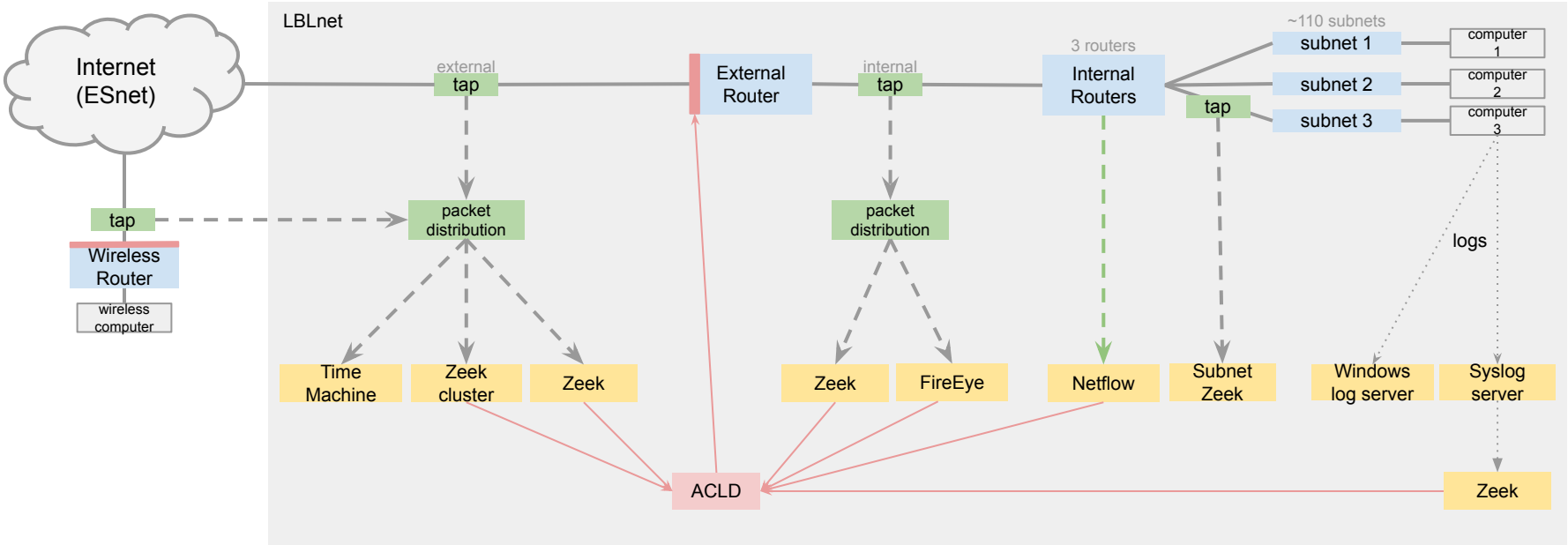
Zeek Network Security Monitor

Goals of Zeek Deployment

- Pervasive visibility
- Forensic Value
  - Ability to reconstruct events
- Non security use case
  - Estimation planning (active licenses)
  - Network queries, debugging, troubleshooting
    - IP poachers
    - misconfigured hosts
- Redundancy
  - Zeek itself
  - Complements other monitors (netflows, syslogs etc.)
- Developing new heuristics
- Research
- Zeek development itself

Deploying Zeek …

Cyber Security: Border Access Control and Visibility



Jay Krous, July 30, 2012

This diagram is for illustrative purposes only, additional technical details require a narrative.

Deployment locations

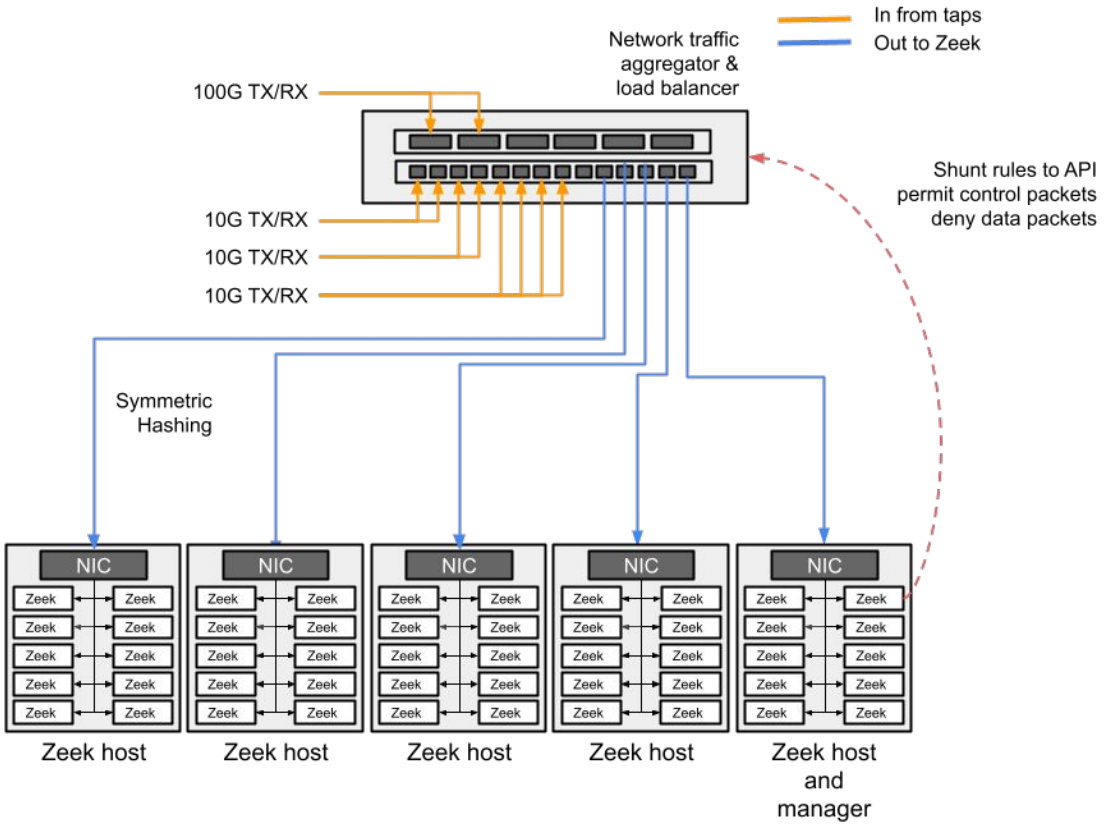
- External DMZ (aka North-South Traffic)
- Internal DMZ (aka north-south Traffic
- Internal Subnets (aka East-West Traffic)
- Strategic locations
  - Behind load balancers, NATs, Firewalls
  - At locations where we can get unencrypted views (database backends)
  - In Front of institutional infrastructure
    - Nameservers, Mail servers, AD, File servers, Central Syslog etc etc.
  - Business Systems

EXT-DMZ and INT-DMZ

EXT-DMZ	INT-DMZ
Ext-DMZ aka Outside the Border router	Inside (Behind?) the Border router
See's ALL the traffic coming in and out of the network	Only sees traffic that is not blocked with Router (ACL, Nullzero, Rules)
Why do we care about connections which are blocked ? ● True situational awareness - visibility ● Who is attacking us/sense of weather outside ● Trends into effectiveness of blocking (1.6Billion connections / day) ● Disks are cheap	● Connections that actually matter ● Faster searches ● Effectiveness of blocking ● (280 Million Connections / day) ● Disks are cheap (so might as well gather this data too)

Commodity Hardware

- We're still a Supermicro shop: cheap, reliable, configurable
- The Game is "Sizing"
  - Finding the balance between the right number of cores vs speed
  - Amount of RAM based on number of workers based on number of interfaces amount of traffic
- NICs: Still have Myricom, but Intel now, still 10G on host
- Clusters configuration options drive division of hardware
  - Multiple separate nodes vs Cluster-in-a-Box



https://go.lbl.gov/100g

The need for the Cores!!

- Historically we've used Intel CPU
- Realized 1 Gbps / worker is a good balance
- Easy scaling with > number of cores to keep up with bandwidth (N x 100Gbps)
- Moving from Intel to AMD EPYC Processors
  - EPYC 9654P (96 Cores) , 9534 (64 Cores), 9454 (48 Cores)

1U Rackmount Chassis 23.5" Deep 17.2" Wide	1
860W Redundant Platinum Power supply 10 x 3.5" Hot-Swap SAS/SATA/NVMe Drive Bays	2
Supermicro H13SSW motherboard 12 x DDR5 4800 ECC RDIMM Slots 2 x PCIe5.0 (x16) slots (FH / HL) 2 x PCIe5.0 (x16) AIOM Slots 2 x USB 3.0 Ports (Rear) 1 x VGA Port 1 x COM Port 1x Dedicated IPMI LAN port	1
1U Mounting Rails	1
Options	
Processors (AMD EPYC) (Max = 1) AMD EPYC 9654P Processor (2.40GHz, 384MB L3, 96C, 360W)	1
DDR5 RDIMM (Max = 12) 32GB DDR5 4800 MT/s ECC RDIMM	12
Drives (Max = 10) Kioxia CM7-R Series (3.84TB 2.5in NVMe PCIe5.0x4)	4
AIOM Slot (Max = 2) Quad Port 10Gb Ethernet (SFP+) (Intel XL710) AIOM Card	1
TPM Security TPM 2.0	1
Operating Systems Rocky Linux 9	1
Support 3-YEAR GLOBAL HELP DESK & LOGISTIC MGMT, KYHD	1
Configuration Notes	

Zeek Monitors: Network Security

One of the foundational elements of setup is the network configuration:

- **Dedicated Subnet with Firewall Protection**
  - Zeek operates within a subnet that is heavily protected by firewalls, ensuring that only authorized traffic is permitted
- **Bastion Hosts for Access**
  - To log in to Zeek, users must go through bastion hosts, adding an additional layer of security
- **No Default Routes to the Internet**
  - This configuration eliminates the possibility of call-backs working, significantly strengthening our defense.
  - The absence of direct Internet access not only mitigates risks but also ensures that intra-subnet detection can be implemented for enhanced monitoring.

Zeek Monitor: Host-Based Security

Host-level configurations are equally critical:

- **Package Management and Updates**
  - Regular updates and package management ensure that the systems remain secure and up-to-date.
- **User Authentication**
  - OTP
  - Dedicated ssh keys
- **Centralized Logging**
  - Logs are aggregated through a centralized syslog system.
  - While logging shell commands is considered, we've disabled history-to-syslog due to associated challenges, acknowledging that this approach has its own trade-offs.

Data Protection: Integrity and Availability of Logs

Protecting integrity of Zeek logs and ensuring availability are highest priorities

- **High integrity ZFS file system storage**
  - Fault tolerant and integrated checksumming with multiple degrees of parity
- **Petabyte Scale "Self-Documenting Flat File Archive"**
  - Easy to find logs or pcaps by content name/file type and date hierarchies
- **Read-Only Permissions and Immutable Flags**:
  - Archives are secured with read-only permissions and the system immutable (‘schg’) flag.
  - However, we recognize the need for standardized practices to *ensure these settings are consistently applied post-compression*.
  - Archives are owned by a different user and read only permissions prevent accidental tmp files etc
- **Redundancy**
  - Duplicate archives maintained on multiple servers in different physical locations
  - Extra "copies" within each file system via hourly/daily/weekly ZFS Snapshots
- **Old-School Fire Safe Vault for Physical Archives on-site**
  - Physical copies of Zeek logs are are securely stored in Vaults
- **TODO List**
  - Richer "metadata" library beyond "self-documenting" hierarchy: tap/sensor histories, per file checksums history, outage/gap tracking
  - Glacier Cloud Archive (Earthquake defense, but needs trustworthy crypto layer)

Source Control and Package Management

Source control and package management are critical to maintaining a secure and stable deployment:

- **Locked-Down Repositories**
  - Access to repositories is tightly controlled to prevent unauthorized changes.
    - Negatives - Zeek packages is somewhat a pain
- **Building Custom Packages**
  - For FreeBSD, we build our own packages using Poudriere, ensuring full control over the software stack.
  - For Linux, we rely on trusted world repositories but mitigate potential risks by comparing behaviors and logs between FreeBSD and Linux systems.
- **Diversity in software Stack**
  - This dual-platform approach provides diversity and redundancy, enhancing our ability to detect anomalies and maintain stability.
  - Approaches of using different operating systems (Linux vs FreeBSD, using different capture cards does allow for some redundancy).

Redundancy in Deployment

- Tap locations
  - Internal DMZ / External DMZ (at least two of each)
- Clusters vs standalone
- Operating system
  - Linux vs FreeBSD
- Capture Cards
  - Intel vs myricom
- Physical architecture
  - Cluster-in-a-box vs multi-node clusters
- Different Taps and traffic aggregators

Monitoring

Continuous monitoring is essential to detect and address issues promptly:

- **Puppet for System Configuration Management**
  - OS environments, users, tap configuration, tuning
  - We build Zeek from source individually for "best fit/custom". Rolled our own zeek package for a while but that got tedious at our scale (tens of zeek hosts, not hundreds)

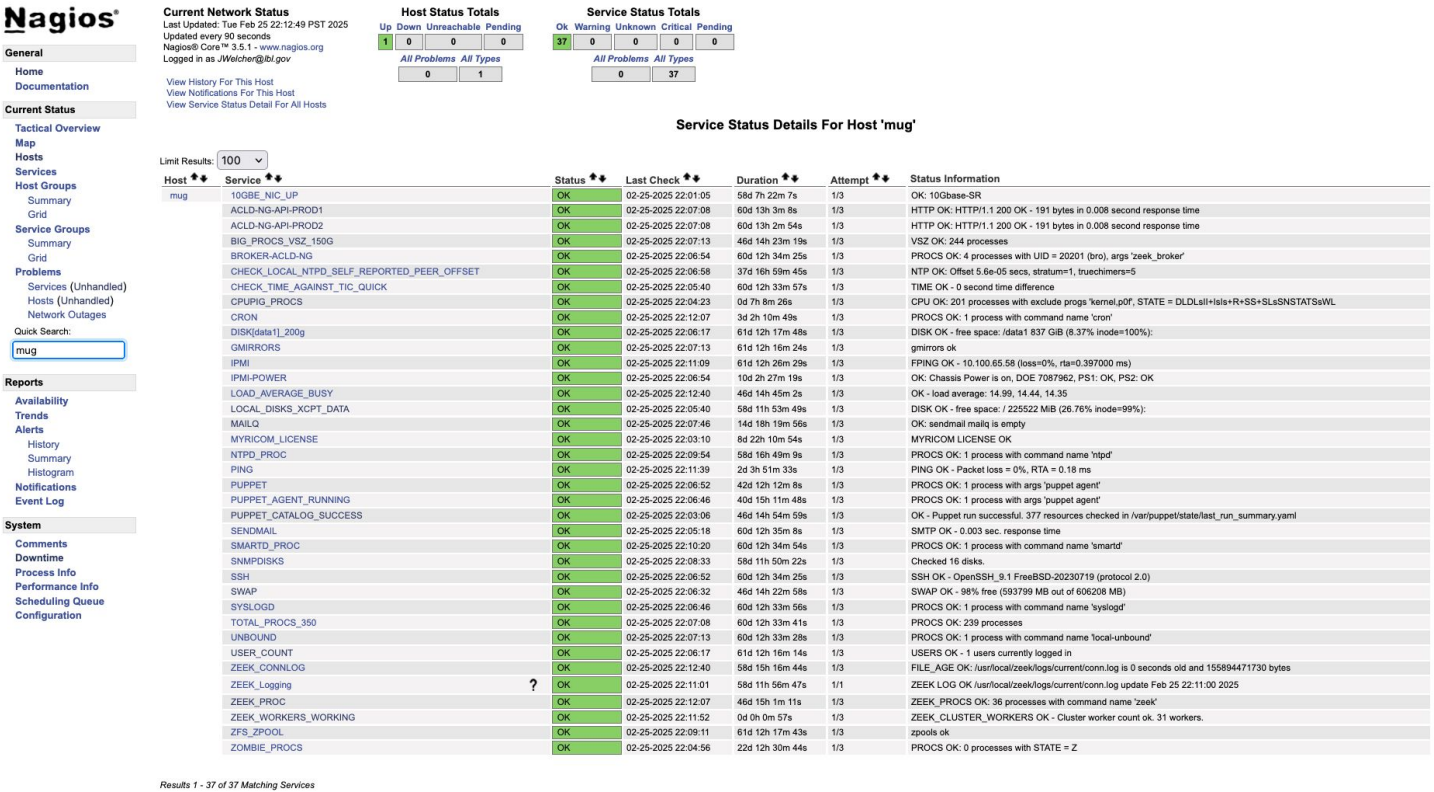
- **Nagios for Files and Processes**
  - Nagios for real-time alerting and notification
  - Check on files, processes, memory, filesystems, time, etc.

- **Grafana for Visualization**
  - Trends and anomaly detection
  - Powerful, still playing with it

Nagios for Files and Processes

- System Monitoring: Useful checks you are probably already running
  - SWAP check if Zeek RAM usage blows up due to policy or weird traffic change
  - CPU check if a worker pegs a core at 100%
  - Network Interface / TAP check: did you lose link (for whatever reason?)
  - (and Puppet continuous system checks is a good backup)
- The Basics Checks
  - `(Processes: Are the right number of `zeek` process running? (should be a stable number)`
  - FILE Age check: Is the conn log less than 1 second old?
  - FILE Size check (min & max! Like when NTP DOS made conn log blow up!)
- Specific checks for that Zeek host's purpose
  - Other Log monitoring (easy to clone stamp conn.log check for SMTP or whatever
- A harder detection: *PARTIAL FAILURES*: Zeek Workers Working
  - What if some workers freeze? Or only one of several taps fails? Logs/procs are OK
  - Checks to see if all of the workers are producing output in the most recent conn log lines
  - e.g. (head -8 $conn; tail -10000 $conn) | zeek-cut peer | sort | uniq -c
- Things we always talk about
  - anomaly detection/deviations from 'norm'; parallel log comparisons; capture loss

Nagios Example



Nagios®

General
- Home
- Documentation

Current Status
- Tactical Overview
- Map
- Hosts
- Services
- Host Groups
  - Summary
  - Grid
- Service Groups
  - Summary
  - Grid
- Problems
  - Services (Unhandled)
  - Hosts (Unhandled)
  - Network Outages

Quick Search: mug

Reports
- Availability
- Trends
- Alerts
  - History
  - Summary
  - Histogram
- Notifications
- Event Log

System
- Comments
- Downtime
- Process Info
- Performance Info
- Scheduling Queue
- Configuration

Current Network Status
Last Updated: Tue Feb 25 22:12:49 PST 2025
Updated every 90 seconds
Nagios® Core™ 3.5.1 - www.nagios.org
Logged in as *JWelcher@lbl.gov*

View History For This Host
View Notifications For This Host
View Service Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
1	0	0	0

All Problems	All Types
0	1

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
37	0	0	0	0

All Problems	All Types
0	37

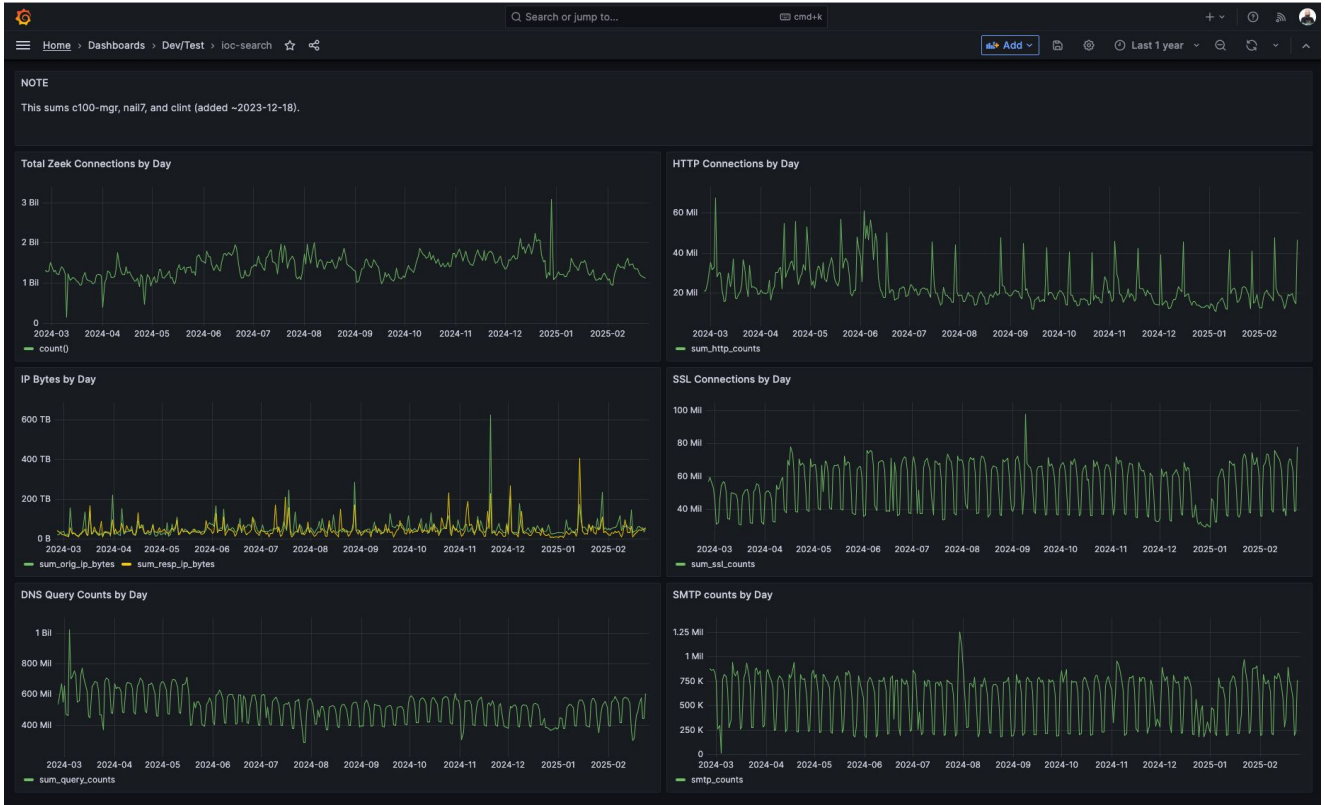
Service Status Details For Host 'mug'

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
mug	10GBE_NIC_UP	OK	02-25-2025 22:01:05	58d 7h 22m 7s	1/3	OK: 10Gbase-SR
	ACLD-NG-API-PROD1	OK	02-25-2025 22:07:08	60d 13h 3m 8s	1/3	HTTP OK: HTTP/1.1 200 OK - 191 bytes in 0.008 second response time
	ACLD-NG-API-PROD2	OK	02-25-2025 22:07:08	60d 13h 2m 54s	1/3	HTTP OK: HTTP/1.1 200 OK - 191 bytes in 0.008 second response time
	BIG_PROCS_VSZ_150G	OK	02-25-2025 22:07:13	46d 14h 23m 19s	1/3	VSZ OK: 244 processes
	BROKER-ACLD-NG	OK	02-25-2025 22:06:54	60d 12h 34m 25s	1/3	PROCS OK: 4 processes with UID = 20201 (bro), args 'zeek_broker'
	CHECK_LOCAL_NTPD_SELF_REPORTED_PEER_OFFSET	OK	02-25-2025 22:06:58	37d 16h 59m 45s	1/3	NTP OK: Offset 5.6e-05 secs, stratum=1, truechimers=5
	CHECK_TIME_AGAINST_TIC_QUICK	OK	02-25-2025 22:05:40	60d 12h 33m 57s	1/3	TIME OK - 0 second time difference
	CPUPIG_PROCS	OK	02-25-2025 22:04:23	0d 7h 8m 26s	1/3	CPU OK: 201 processes with exclude progs 'kernel,p0f', STATE = DLDLsIl+lsls+R+SS+SLsSNSTATSsWL
	CRON	OK	02-25-2025 22:12:07	3d 2h 10m 49s	1/3	PROCS OK: 1 process with command name 'cron'
	DISK[data1]_200g	OK	02-25-2025 22:06:17	61d 12h 17m 48s	1/3	DISK OK - free space: /data1 837 GiB (8.37% inode=100%):
	GMIRRORS	OK	02-25-2025 22:07:13	61d 12h 16m 24s	1/3	gmirrors ok
	IPMI	OK	02-25-2025 22:11:09	61d 12h 26m 29s	1/3	FPING OK - 10.100.65.58 (loss=0%, rta=0.397000 ms)
	IPMI-POWER	OK	02-25-2025 22:06:54	10d 2h 27m 19s	1/3	OK: Chassis Power is on, DOE 7087962, PS1: OK, PS2: OK
	LOAD_AVERAGE_BUSY	OK	02-25-2025 22:12:40	46d 14h 45m 2s	1/3	OK - load average: 14.99, 14.44, 14.35
	LOCAL_DISKS_XCPT_DATA	OK	02-25-2025 22:05:40	58d 11h 53m 49s	1/3	DISK OK - free space: / 225522 MiB (26.76% inode=99%):
	MAILQ	OK	02-25-2025 22:07:46	14d 18h 19m 56s	1/3	OK: sendmail mailq is empty
	MYRICOM_LICENSE	OK	02-25-2025 22:03:10	8d 22h 10m 54s	1/3	MYRICOM LICENSE OK
	NTPD_PROC	OK	02-25-2025 22:09:54	58d 16h 49m 9s	1/3	PROCS OK: 1 process with command name 'ntpd'
	PING	OK	02-25-2025 22:11:39	2d 3h 51m 33s	1/3	PING OK - Packet loss = 0%, RTA = 0.18 ms
	PUPPET	OK	02-25-2025 22:06:52	42d 12h 12m 8s	1/3	PROCS OK: 1 process with args 'puppet agent'
	PUPPET_AGENT_RUNNING	OK	02-25-2025 22:06:46	40d 15h 11m 48s	1/3	PROCS OK: 1 process with args 'puppet agent'
	PUPPET_CATALOG_SUCCESS	OK	02-25-2025 22:03:06	46d 14h 54m 59s	1/3	OK - Puppet run successful. 377 resources checked in /var/puppet/state/last_run_summary.yaml
	SENDMAIL	OK	02-25-2025 22:05:18	60d 12h 35m 8s	1/3	SMTP OK - 0.003 sec. response time
	SMARTD_PROC	OK	02-25-2025 22:10:20	60d 12h 34m 54s	1/3	PROCS OK: 1 process with command name 'smartd'
	SNMPDISKS	OK	02-25-2025 22:08:33	58d 11h 50m 22s	1/3	Checked 16 disks.
	SSH	OK	02-25-2025 22:06:52	60d 12h 34m 25s	1/3	SSH OK - OpenSSH_9.1 FreeBSD-20230719 (protocol 2.0)
	SWAP	OK	02-25-2025 22:06:32	46d 14h 22m 58s	1/3	SWAP OK - 98% free (593799 MB out of 606208 MB)
	SYSLOGD	OK	02-25-2025 22:06:46	60d 12h 33m 56s	1/3	PROCS OK: 1 process with command name 'syslogd'
	TOTAL_PROCS_350	OK	02-25-2025 22:07:08	60d 12h 33m 41s	1/3	PROCS OK: 239 processes
	UNBOUND	OK	02-25-2025 22:07:13	60d 12h 33m 28s	1/3	PROCS OK: 1 process with command name 'local-unbound'
	USER_COUNT	OK	02-25-2025 22:06:17	61d 12h 16m 14s	1/3	USERS OK - 1 users currently logged in
	ZEEK_CONNLOG	OK	02-25-2025 22:12:40	58d 15h 16m 44s	1/3	FILE_AGE OK: /usr/local/zeek/logs/current/conn.log is 0 seconds old and 155894471730 bytes
	ZEEK_Logging ?	OK	02-25-2025 22:11:01	58d 11h 56m 47s	1/1	ZEEK LOG OK /usr/local/zeek/logs/current/conn.log update Feb 25 22:11:00 2025
	ZEEK_PROC	OK	02-25-2025 22:12:07	46d 15h 1m 11s	1/3	ZEEK_PROCS OK: 36 processes with command name 'zeek'
	ZEEK_WORKERS_WORKING	OK	02-25-2025 22:11:52	0d 0h 0m 57s	1/3	ZEEK_CLUSTER_WORKERS OK - Cluster worker count ok. 31 workers.
	ZFS_ZPOOL	OK	02-25-2025 22:09:11	61d 12h 17m 43s	1/3	zpools ok
	ZOMBIE_PROCS	OK	02-25-2025 22:04:56	22d 12h 30m 44s	1/3	PROCS OK: 0 processes with STATE = Z

Results 1 - 37 of 37 Matching Services

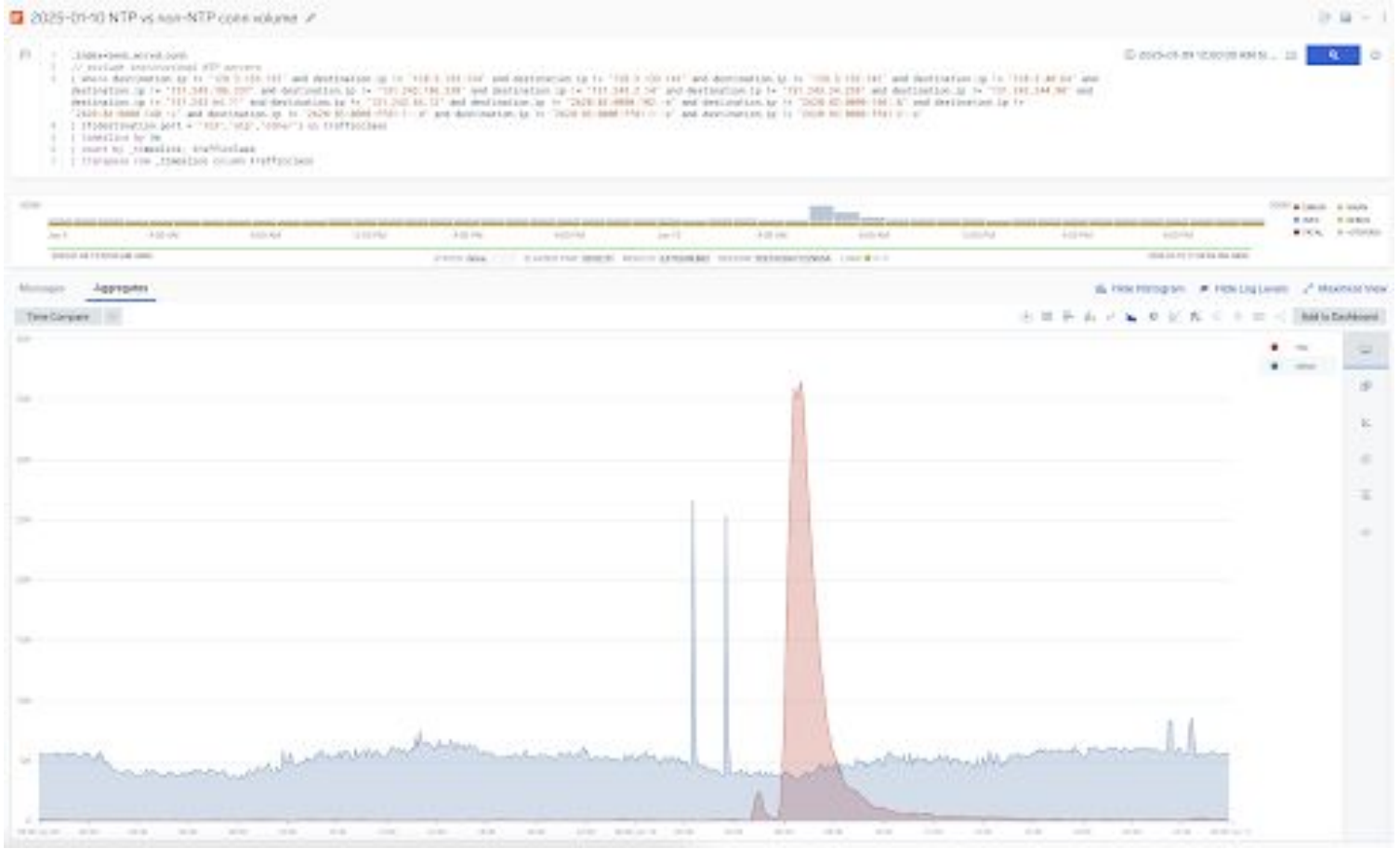
Grafana Zeek Log Visualization



Scaling for the logs

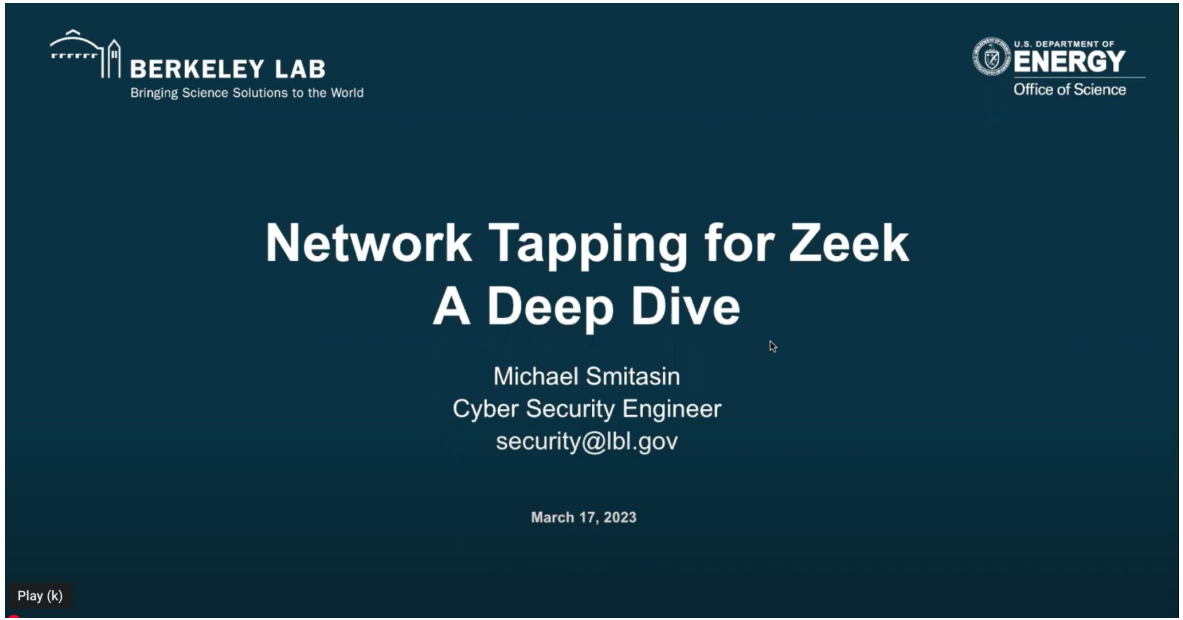
- If we can search fast we will search more
- Zeek logs in plain text
  - GNU parallel work wonders but need cores and still somewhat slow (180 core boxes)
- Clickhouse
  - As of now 1000+ days can be search in seconds (1.6 Billion conn.logs / day or about 200Gb)
- Sumologic and advantage
  - Fast searching on infinite archive
  - Alerting on the logs helps significantly

Sumologic show casing the NTP DoS from .KR



Deploying Zeek : Tapping infrastructure

- That’s a talk of its Own: https://www.youtube.com/watch?v=VIkmin2kpLk
- Stay tuned for next update - we are working on Tap infra upgrades



In conclusion, 5 things to take away …

- Firewalled off, L4 basion protected sensors
- No default route on sensors
- Diverse and redundant deployment on different OSes/architectures/hardware, etc.
- Unlikely any one particular "hack" could affect all sensors (though we have seen weird network traffic tickle bugs in multiple sensors, but never all of them.).
- Diversity in releases of Zeek is also a strength.

Questions

asharma@lbl.gov

jwelcher@lbl.gov

security@lbl.gov

Happy to share our scripts, monitoring and other configurations with you !!

We use Zeek! You should too ….