

DCC 2013

# **Introduction to SIE and 2013 Update**

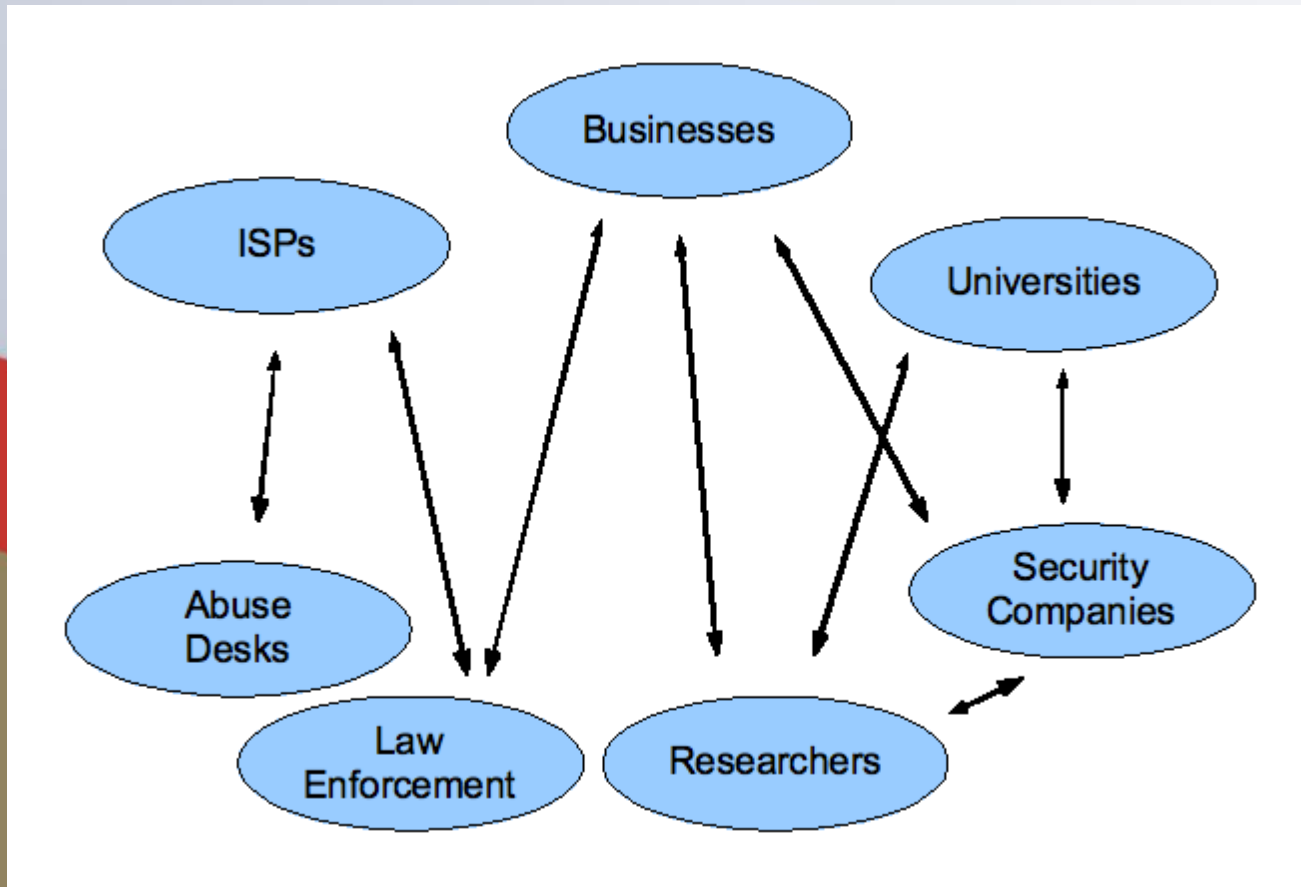
**Paul Vixie  
ISC**



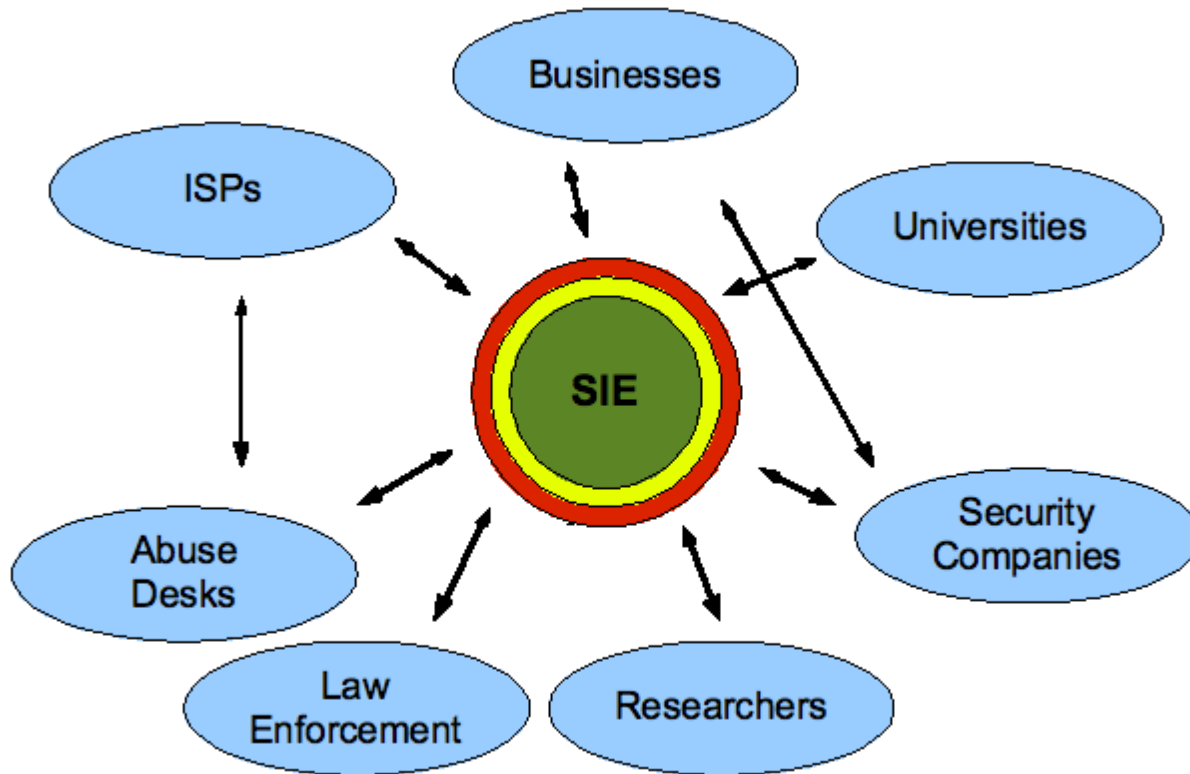
# Security Information Exchange?

- Old: exclusive sensor/analyst “silos”
- New: compete on execution not data
- Goal: everybody’s customers get safer
- Method: mix of public/private channels
- Focus: real time ( $\sim$ DSP), not “batch”
- Motivation: ISC is also an analyst
- Cost: everybody pays what they can afford, often a mix of cash and data

# Decentralized - bi-lateral

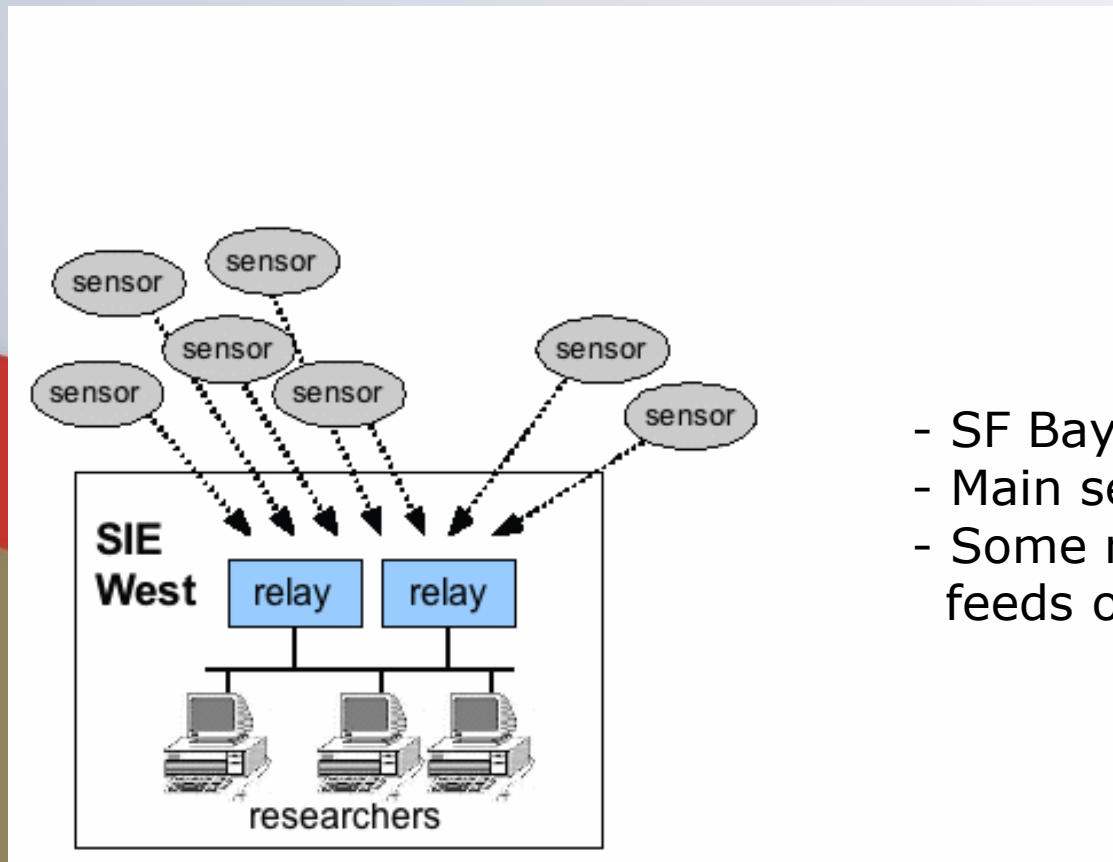


# Centralized - multi-lateral



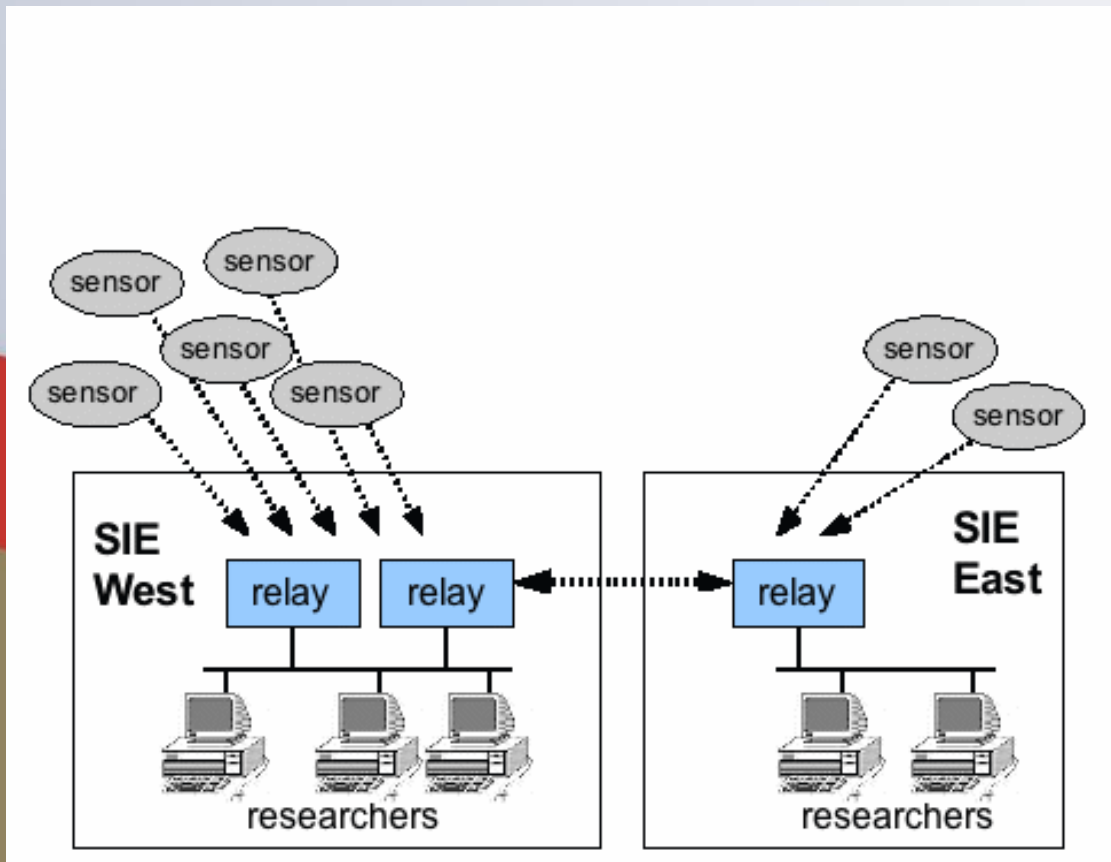
Efficient **sharing** within common **legal/privacy** framework

# Data distribution model - original



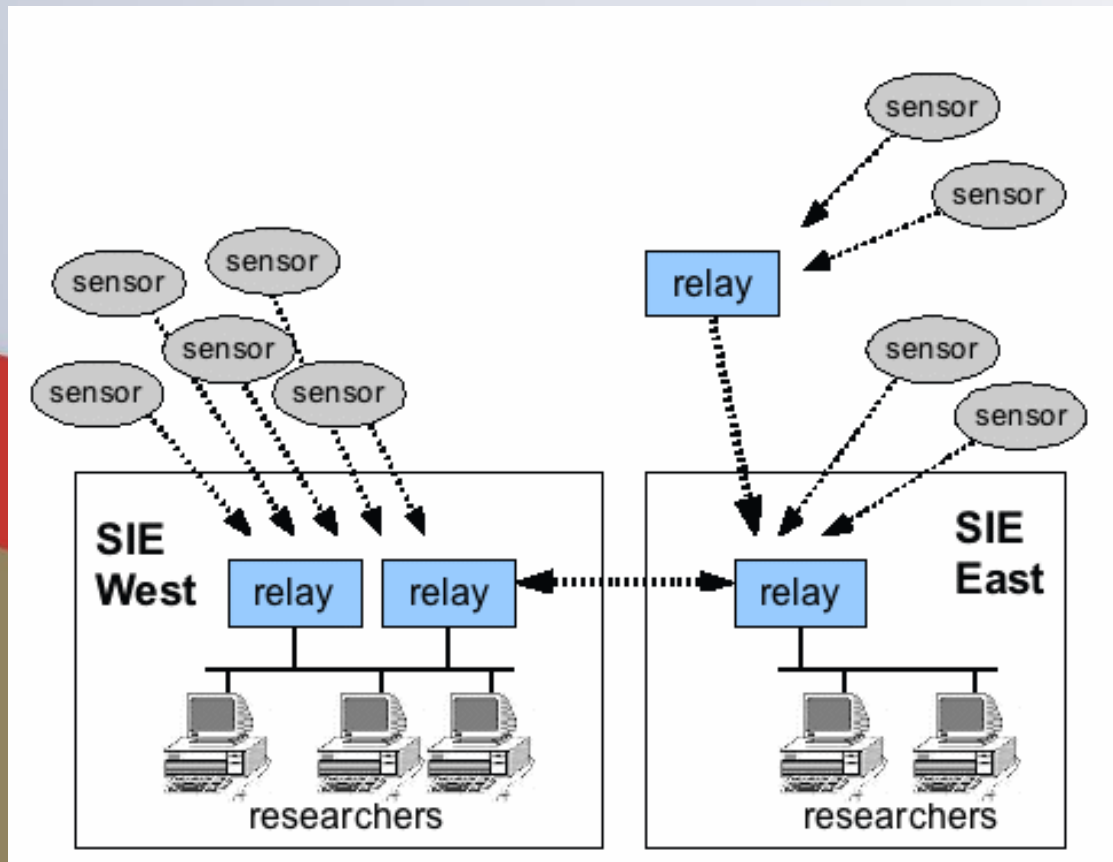
- SF Bay Area, US (PAIX)
- Main sensor relays
- Some researchers getting feeds off switches

# Data distribution model – east++



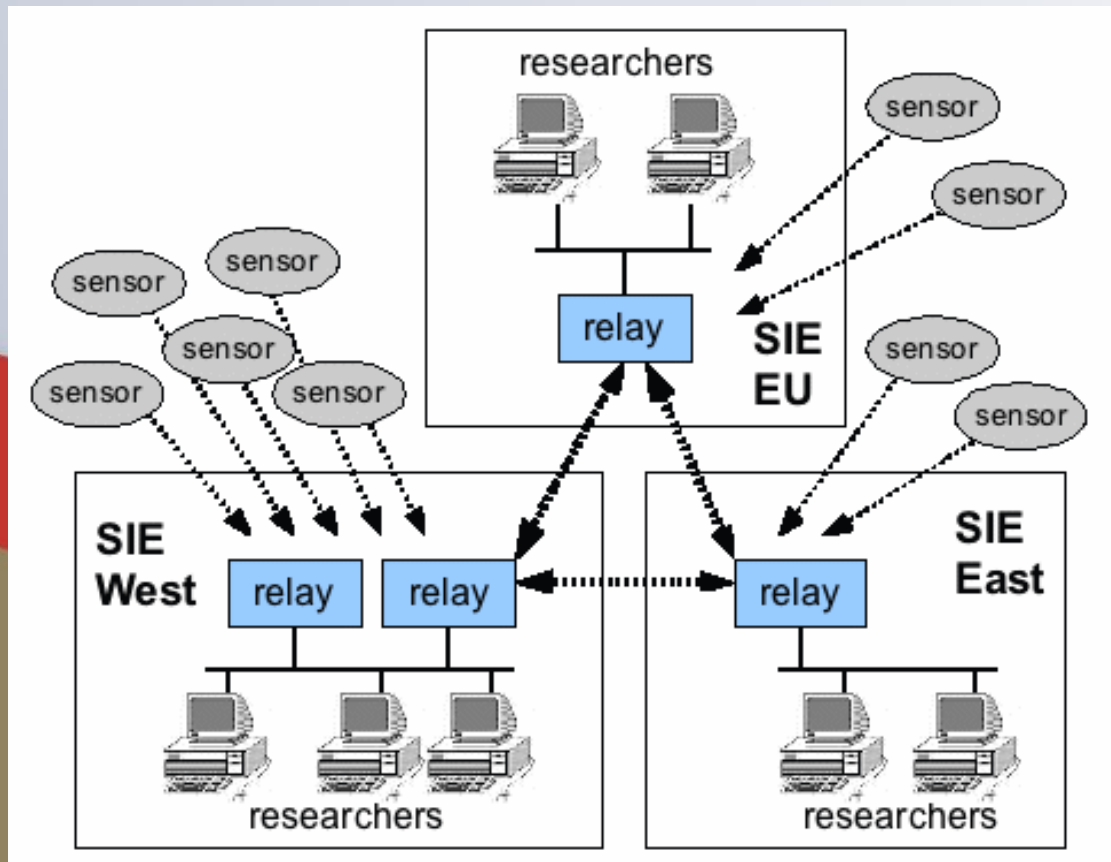
- DC or NY, US
- Redundant facilities
- More researchers

# Data distribution model - relay



- Add relays at exchanges in different countries
- Add local sensors
- Local sharing or tools possible within relay

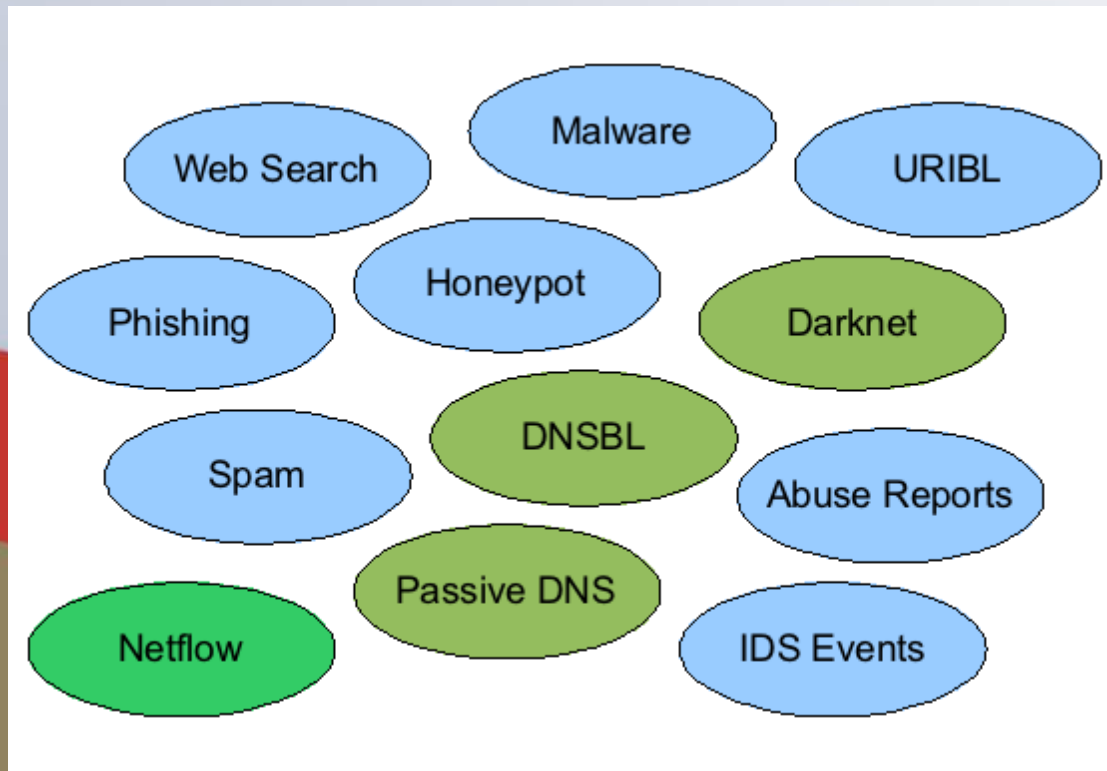
# Data distribution model - future



- Promote node when number of researchers is significant
- Scaling issues

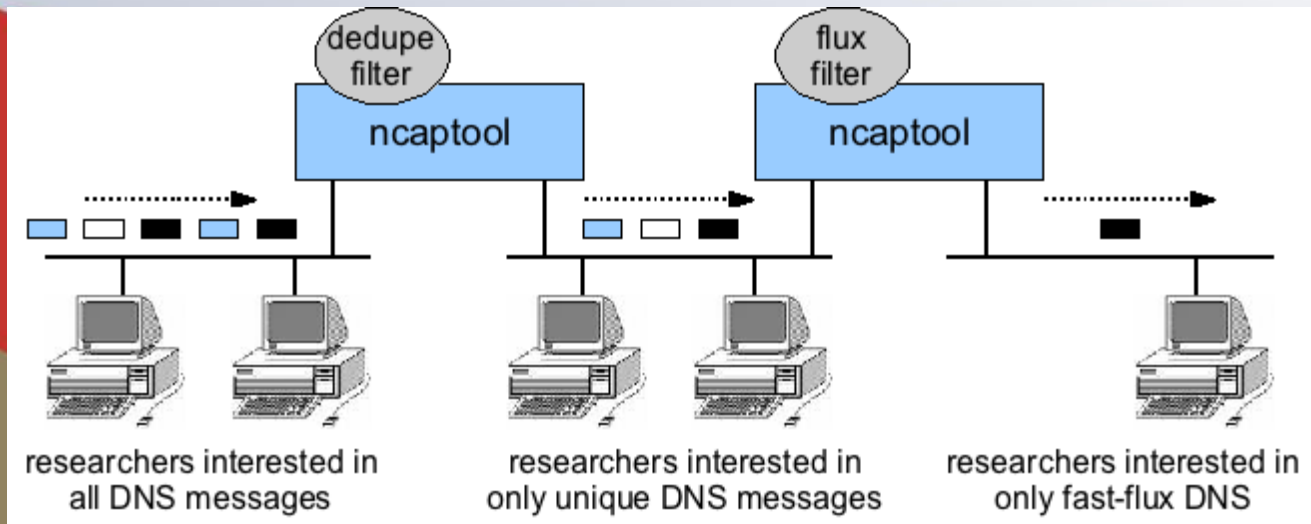


# Disparate data



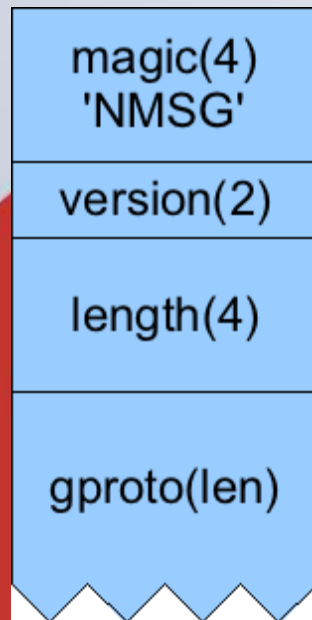
# ncap

- plug-in filters in action



# nmsg

- <ftp://ftp.isc.org/isc/nmsg>
- Any structured data
- Format:



# Google protocol buffers

- <http://code.google.com/apis/protocolbuffers>
- APIs for C, C++, Python, Java, Perl
- Arguably better than XML
- Why reinvent the wheel?
- Open source
- Extensible:

# nmsg.proto

```
package nmsg;
```

```
message Nmsg {  
    repeated NmsgPayload  payloads = 1;  
}
```

```
message NmsgFragment {  
    required uint32  id = 1;  
    required uint32  current = 2;  
    required uint32  last = 3;  
    required bytes   fragment = 4;  
}
```

```
message NmsgPayload {  
    required int32    vid = 1;  
    required int32    msgtype = 2;  
    required int64    time_sec = 3;  
    required fixed32  time_nsec = 4;  
    optional bytes    payload = 5;  
    repeated uint32   user = 6;  
}
```

# isc/email.proto

```
package nmsg.isc;
```

```
enum EmailType {  
    unknown = 0;  
    spamtrap = 1;    // email sent to a spamtrap  
    rej_network = 2; // rejected by network or SMTP (pre-DATA) checks  
    rej_content = 3; // rejected by content filter (including domain blacklists)  
    rej_user = 4;    // classified by user as spam  
}
```

```
message Email {  
    optional EmailType type = 8;  
    optional bytes    headers = 2; // SMTP headers  
    optional bytes    srcip = 3;    // remote client IP  
    optional bytes    srchost = 4;  // remote client PTR, if known  
    optional bytes    helo = 5;    // HELO/EHLO parameter  
    optional bytes    from = 6;    // MAIL FROM parameter (brackets stripped)  
    repeated bytes    rcpt = 7;    // RCPT TO parameter(s) (brackets stripped)  
    repeated bytes    bodyurl = 9; // URL(s) found in decoded body  
}
```

# Conficker Sinkhole Example

- We generate the DNS content and run the DNS servers; instrument w/ NMSG
- We run the HTTP → NMSG servers
- These NMSG flows are relayed into SIE
- The CWG server copies these files into per-analyst directories for use w/ rsync
- SIE-connected analysts get it real time

# Ghost Click Example

- We ran the replacement DNS servers, instrumented with NMSG (queries only)
- Batched for up to one minute to save bandwidth and allow for encryption
- SIE analysts got the data immediately
- We stored copies of the NMSG as files
- Others had to periodically poll (rsync)



# Darknet Example

- Sometimes called “network telescope”
- IP space known in BGP but not used
- Internet cosmic background radiation
- Exception to our “always NMSG” rule
- Any BGP speaking router can be a sensor: replace Null0 with GRE0
- We need more/smaller sensors (many)

# Important Takeaways, SIE/NMSG

- It's all real time, but files can be made
- Network of private Ethernet switches
- Most analysts provide or rent a server
- ~25 channels today, some private
- ~40 analysts: comm/acad/police
- ~500Mbit/sec today, some reprocessed
- SIE pricing is "nondiscriminatory"
- We want more data and more analysts

# Motivation to Participate

- Operator: run a single kind of sensor, let us deliver to all qualified parties
- Analyst: receive a firehose of real time data in a simple binary format
- Us: offer cash discount on services to analysts who can bring data (+DNSDB)
- Economy: lowers total cost of visibility, aligns individual motives with society's
- "Snowball effect"

# Comparison to *uSoft* DCU's SaaS

- Sinkhole as a Service (SaaS)
- Capture botnet C&C, parse all "hits"
- Subscribers are network operators
- Each subscriber provides "Azure" creds
- DCU team populates Azure, many files
- Currently handles ~100m per day
- Compressed text files – easy to use
- DCU data is free; Azure is very cheap

# Questions?

- Email: [info@sie.isc.org](mailto:info@sie.isc.org)
- Web: <https://sie.isc.org/>