High-Performance Network Security Monitoring at the Lawrence Berkeley National Lab

Strategies for Monitoring External and Internal Activity

Robin Sommer

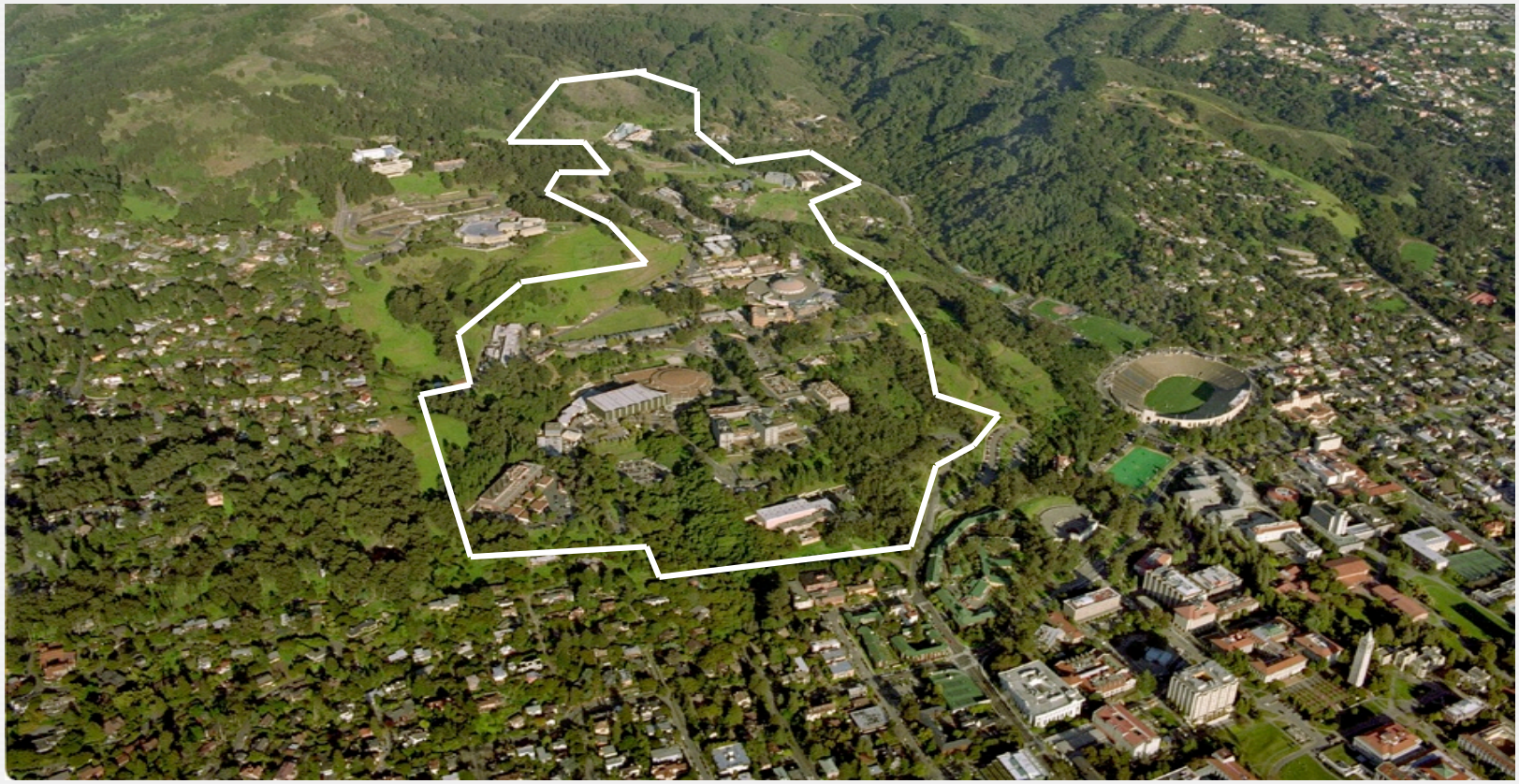
Lawrence Berkeley National Laboratory &
International Computer Science Institute

rsommer@lbl.gov
http://www.icir.org

Fall 2007 Internet2 Member Meeting

Lawrence Berkeley National Lab

- Main site located on a 200-acre area in the Berkeley hills
- Close proximity to UC Berkeley



Lawrence Berkeley National Lab

- Managed by UC for the Department of Energy
- Open, unclassified research
  - Research is freely shared
  - Collaboration around the world
- Diversity of research
  - Nanotechnology, Energy, Physics, Biology, Chemistry, Environmental, Computing
- Diverse user community
  - Scientific facilities used by researchers around the world
  - Many users are transient and not employees
  - Many staff have dual appointment with UC Berkeley
- Very liberal, default-allow security policy
  - Characteristic for many research environment
  - Requires comprehensive approach to monitoring

Network Security Monitoring at the Lab

Outline of this Talk

- Monitoring external activity with the Bro NIDS
  - Overview of Bro's design & architecture, and LBNL's installation
  - Recent Bro Developments
    - The Bro Cluster
    - Dynamic Protocol Detection
- Monitoring internal activity
  - Inter- & intra-subnet monitoring
  - Host-based monitoring
- Outlook

The Bro NIDS

System Philosophy

- Bro is being developed at LBNL & ICSI since 1996
  - LBNL has been using Bro operationally for >10 years
  - It is one of the main components of the lab's network security infrastructure
- Bro provides a real-time network analysis framework
  - Primary a network intrusion detection system (NIDS)
  - However it is also used for pure traffic analysis
- Focus is on
  - Application-level semantic analysis (rather than analyzing individual packets)
  - Tracking information over time
- Strong separation of mechanism and policy
  - The core of the system is policy-neutral (no notion of "good" or "bad")
  - User provides local site policy

System Philosophy (2)

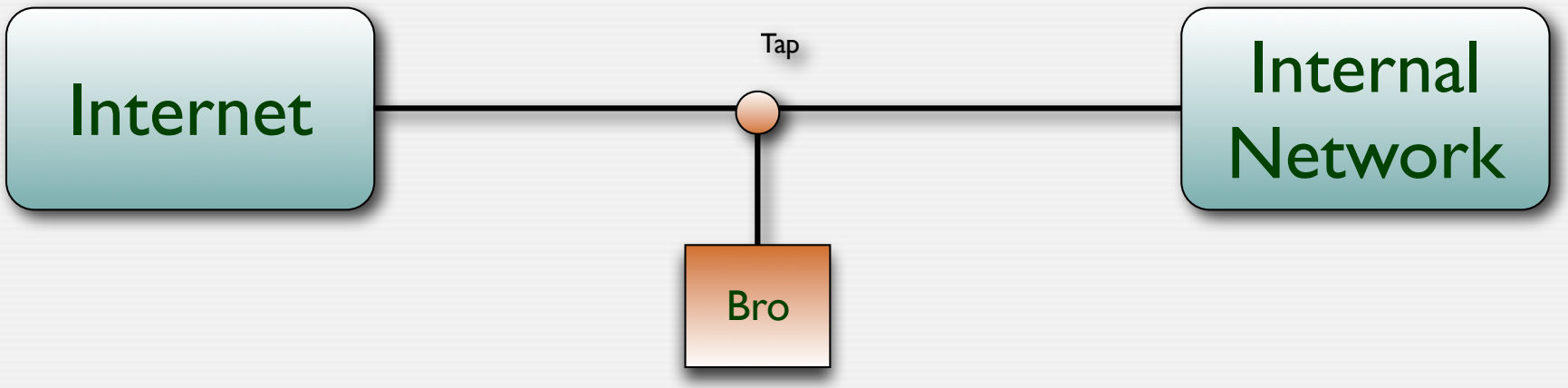
- Operators *program* their policy
  - Not really meaningful to talk about what Bro detects "by default"
- Analysis model is *not* signature matching
  - Bro is fundamentally different from, e.g., Snort (though it *can* do signatures as well)
- Analysis model is *not* anomaly detection
  - Though it does support such approaches (and others) in principle
- System thoroughly logs all activity
  - It does not just alert
  - Logs are invaluable for forensics

Target Environments

- Bro is specifically well-suited for scientific environments
  - Extremely useful in networks with liberal ("default allow") policies
  - High-performance on commodity hardware
  - Supports intrusion prevention schemes
  - Open-source (BSD license)
- It does however require some effort to use effectively
  - Pretty complex, script-based system
  - Requires understanding of the network
  - No GUI, just ASCII logs
  - Only partially documented
  - Lacking resources to fully polish the system
- Development is primarily driven by *research*
  - However, our focus is operational use; we invest much time into "practical" issues
  - Want to bridge gap between research and operational deployment

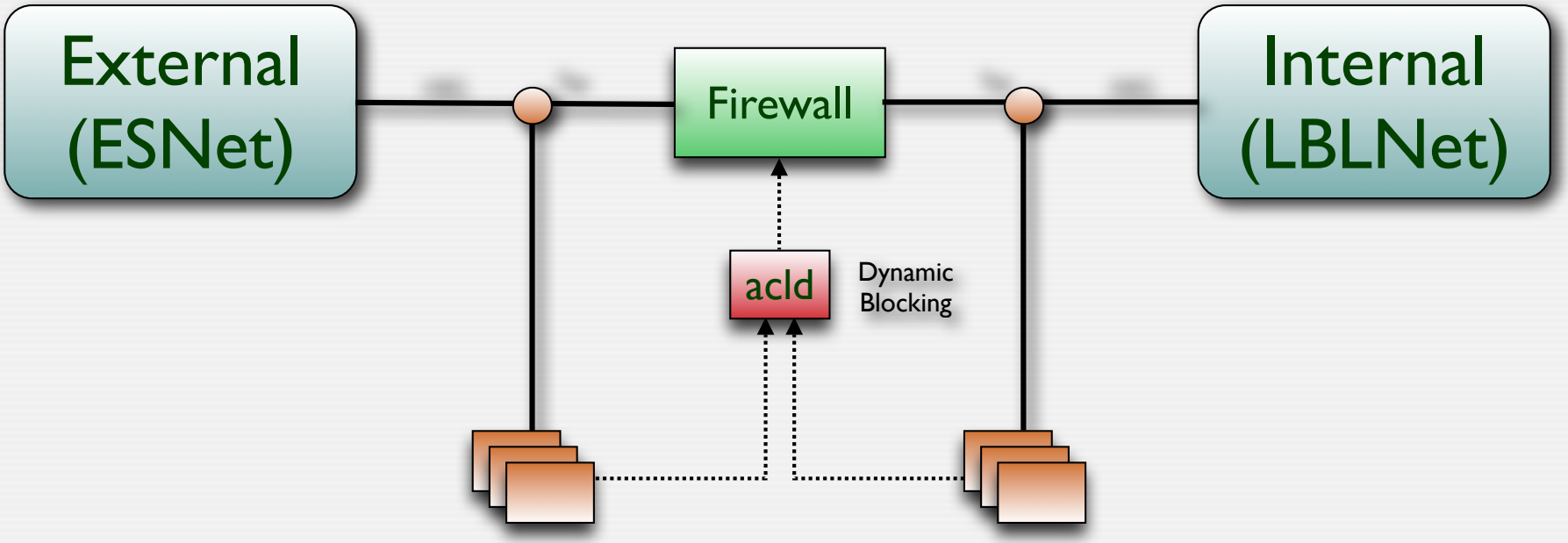
Bro Deployment

- Bro is typically deployed at a site's upstream link
  - Monitors all external packets coming in or going out
  - Deployment similar to other NIDS

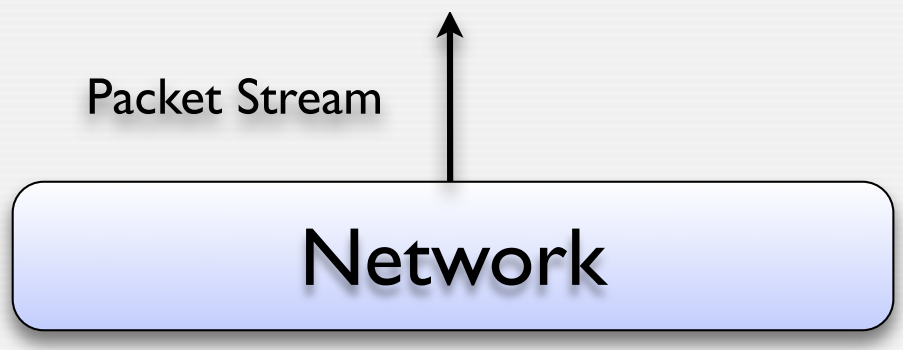


LBNL's Bro Setup

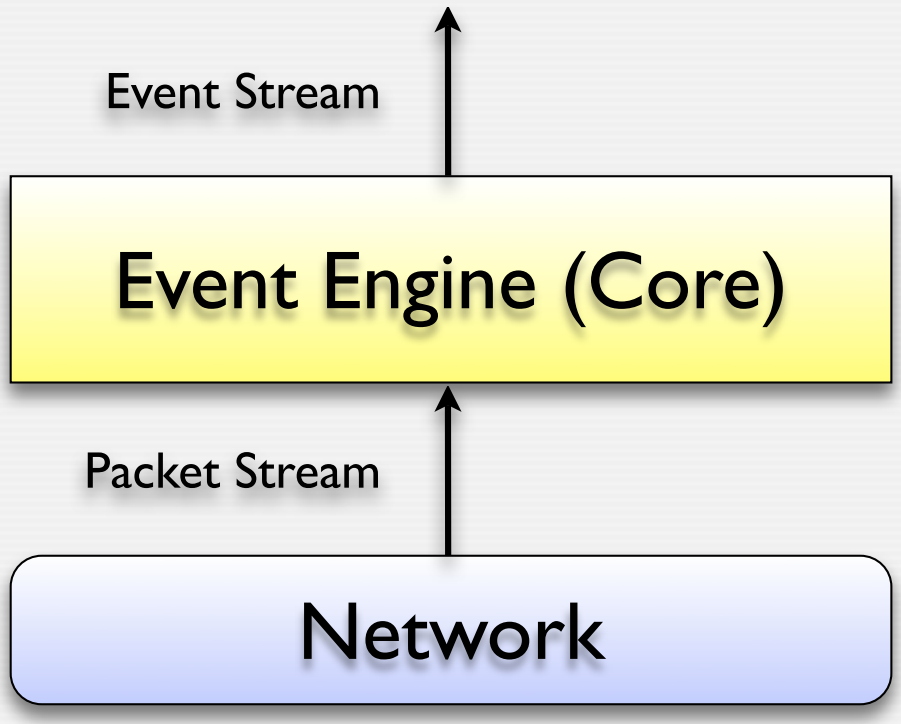
- Uses Bro to monitor its 10 Gbps Internet uplink
  - Several Bro boxes for different tasks, both before & after the firewall
  - Automatically blocks attackers (about 4000 addresses per day!)



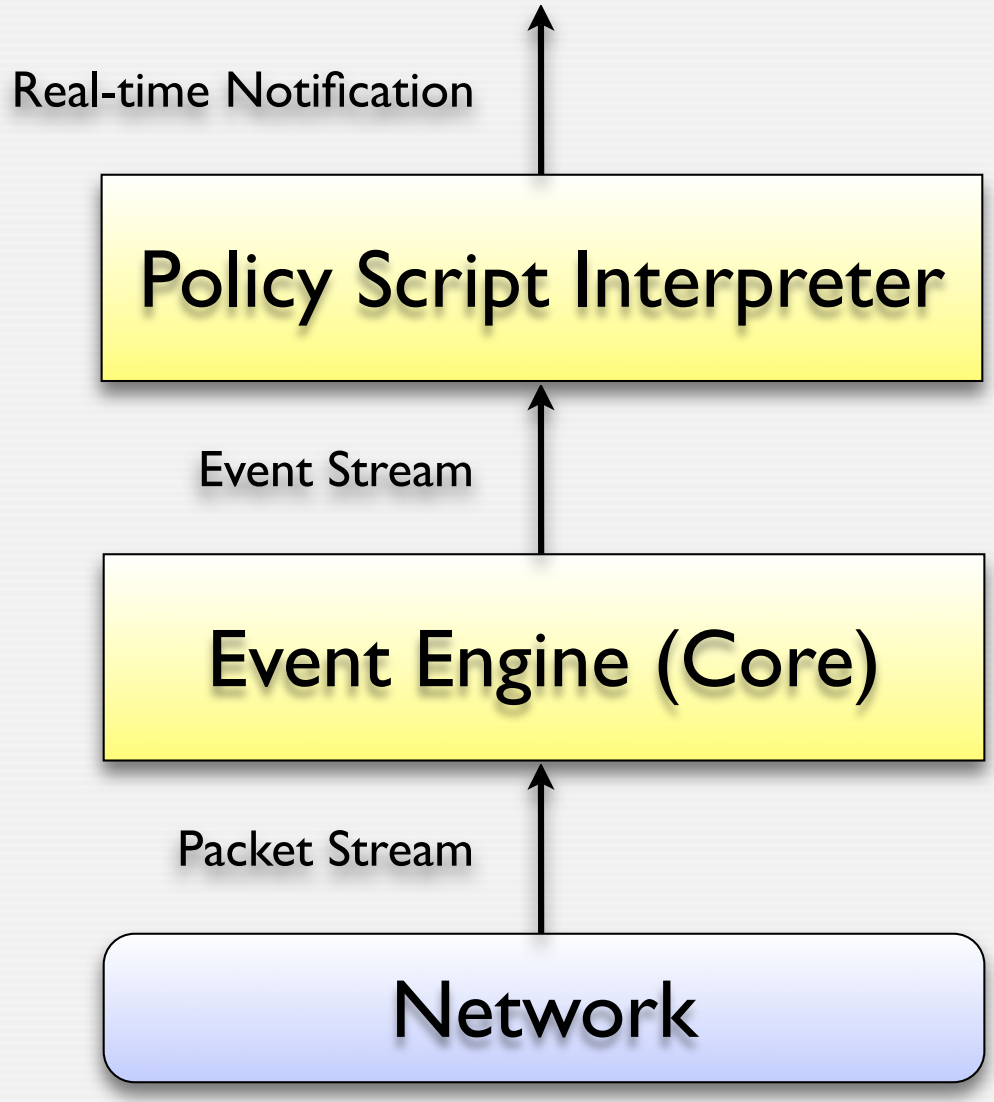
Architecture



Architecture



Architecture



Event-Engine

- Event-engine is written in C++
- Performs *policy-neutral* analysis
  - Turns low-level activity into high-level events
  - Examples: `connection_established, http_request`
  - Events are annotated with context (e.g., IP addresses, URL)
- Contains *analyzers* for >30 protocols, including
  - ARP, IP, ICMP, TCP, UDP
  - DCE-RPC, DNS, FTP, Finger, Gnutella, HTTP, IRC, Ident, NCP, NFS, NTP, NetBIOS, POP3, Portmapper, RPC, Rsh, Rlogin, SMB, SMTP, SSH, SSL, SunRPC, Telnet
- Analyzers generate ~300 types of events

Policy Scripts

- Scripts process event stream, incorporating ...
  - ... context from past events
  - ... site's local security policy
- Scripts take actions
  - Generating alerts via syslog or mail
  - Executing program as a form of response
  - Recording activity to disk

Example Log: Connection Summaries

- One-line summaries for all TCP connections
- Most basic, yet also one of the most useful analyzers

```
> bro -r trace tcp

Time                Duration  Source          Destination
1144876596.658302   1.206521  192.150.186.169 62.26.220.2 \
  http  53052    80       tcp    874       1841      SF     X
  Serv  SrcPort  DstPort  Proto  SrcBytes  DstBytes  State  Dir
```

LBNL has connection logs for every connection attempt since June 94!

Example Log: HTTP Session

```
1144876588.30 %2 start 192.150.186.169:53041 > 195.71.11.67:80
1144876588.30 %2 GET /index.html (200 "OK" [57634] www.spiegel.de)
1144876588.30 %2 > HOST:  www.spiegel.de
1144876588.30 %2 > USER-AGENT:  Mozilla/5.0 (Macintosh; PPC Mac OS ...
1144876588.30 %2 > ACCEPT:  text/xml,application/xml,application/xhtml ...
1144876588.30 %2 > ACCEPT-LANGUAGE:  en-us,en;q=0.7,de;q=0.3
[...]
1144876588.77 %2 < SERVER:  Apache/1.3.26 (Unix) mod_fastcgi/2.2.12
1144876588.77 %2 < CACHE-CONTROL:  max-age=120
1144876588.77 %2 < EXPIRES:  Wed, 12 Apr 2006 21:18:28 GMT
[...]
1144876588.77 %2 <= 1500 bytes: "<!-- Vignette StoryServer 5.0 Wed Apr..."
1144876588.78 %2 <= 1500 bytes: "r "http://spiegel.ivwbox.de" r..."
1144876588.78 %2 <= 1500 bytes: "icon.ico" type="image/ico">^M^J ..."
1144876588.94 %2 <= 1500 bytes: "erver 5.0 Mon Mar 27 15:56:55 ..."
[...]
```

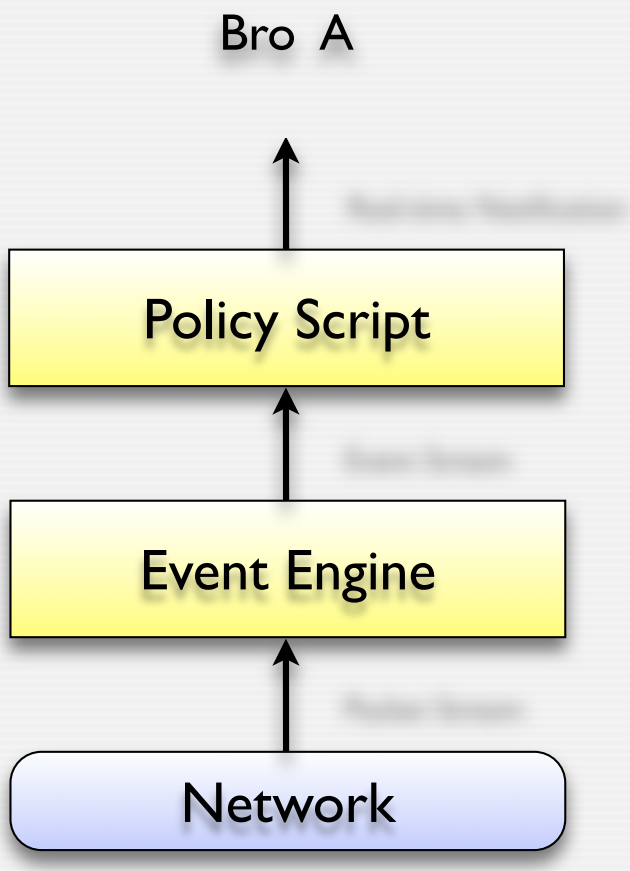
Script Example: Tracking SSH Hosts

```
global ssh_hosts: set[addr];

event connection_established(c: connection)
    {
    local responder = c$id$resp_h; # Responder's address
    local service = c$id$resp_p;   # Responder's port

    if ( service != 22/tcp )
        return; # Not SSH.

    if ( responder in ssh_hosts )
        return; # We already know this one.

    add ssh_hosts[responder]; # Found a new host.
    print "New SSH host found", responder;
    }
```

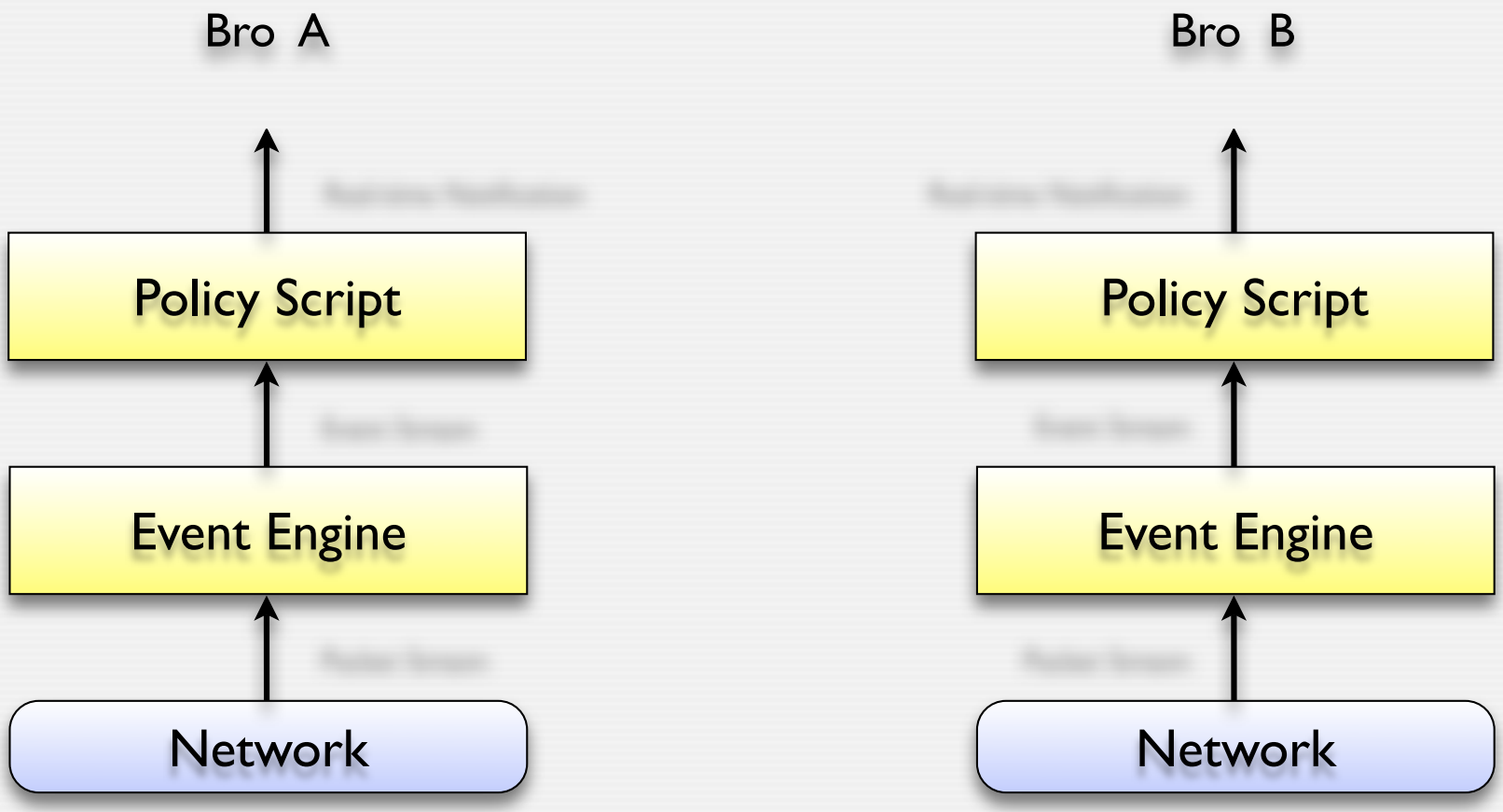
Expressing Policy

- Scripts are written in custom, domain-specific language
  - Bro ships with 20K+ lines of script code
  - Default scripts detect attacks & log activity extensively
- Language is
  - Procedural
  - Event-based
  - Strongly typed
  - Rich in types
    - Usual script-language types, such as tables and sets
    - Domain-specific types, such as addresses, ports, subnets
  - Supporting state management (expiration, timers, etc.)
  - Supporting communication with other Bro instances

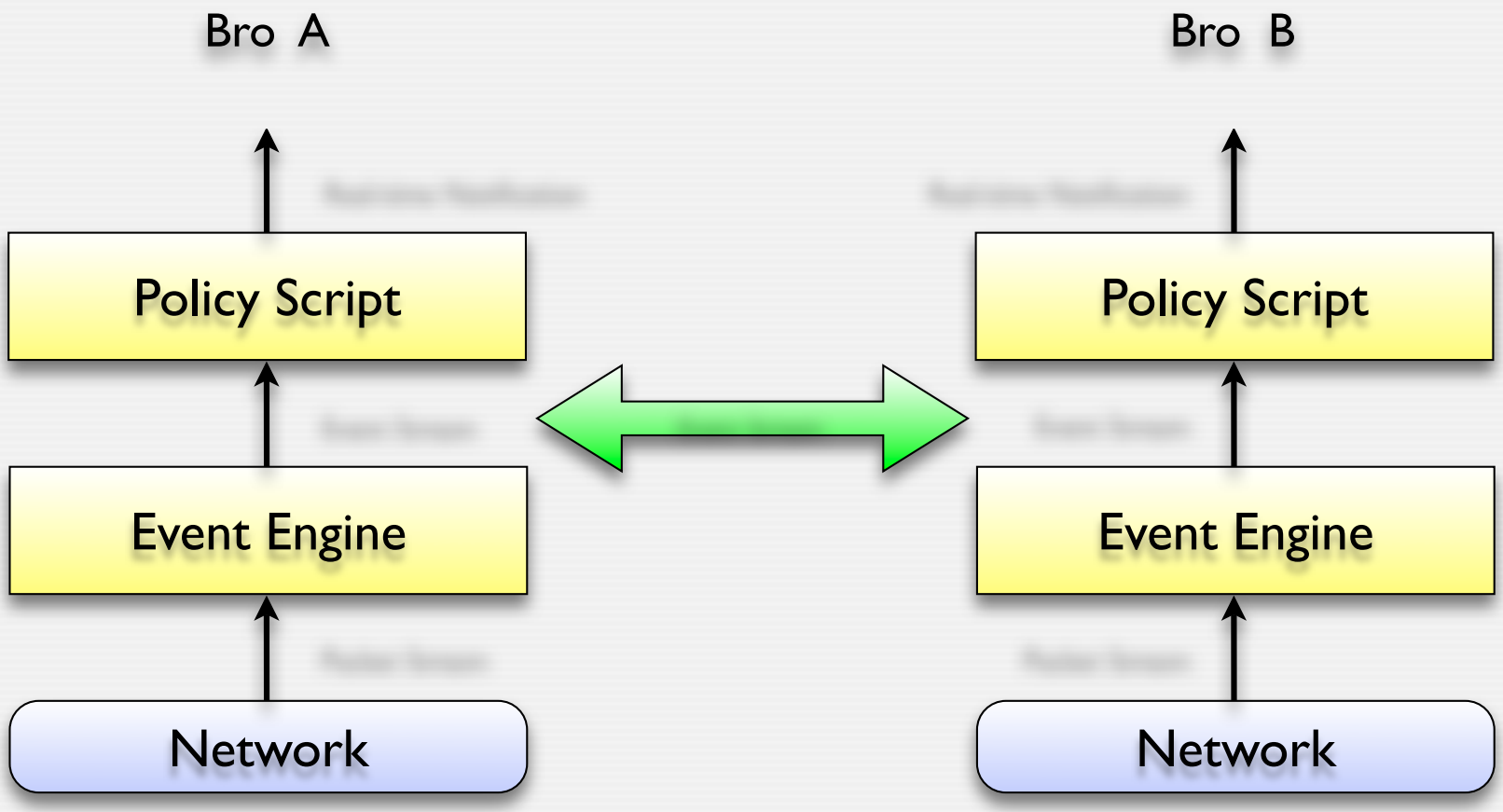
Communication Architecture



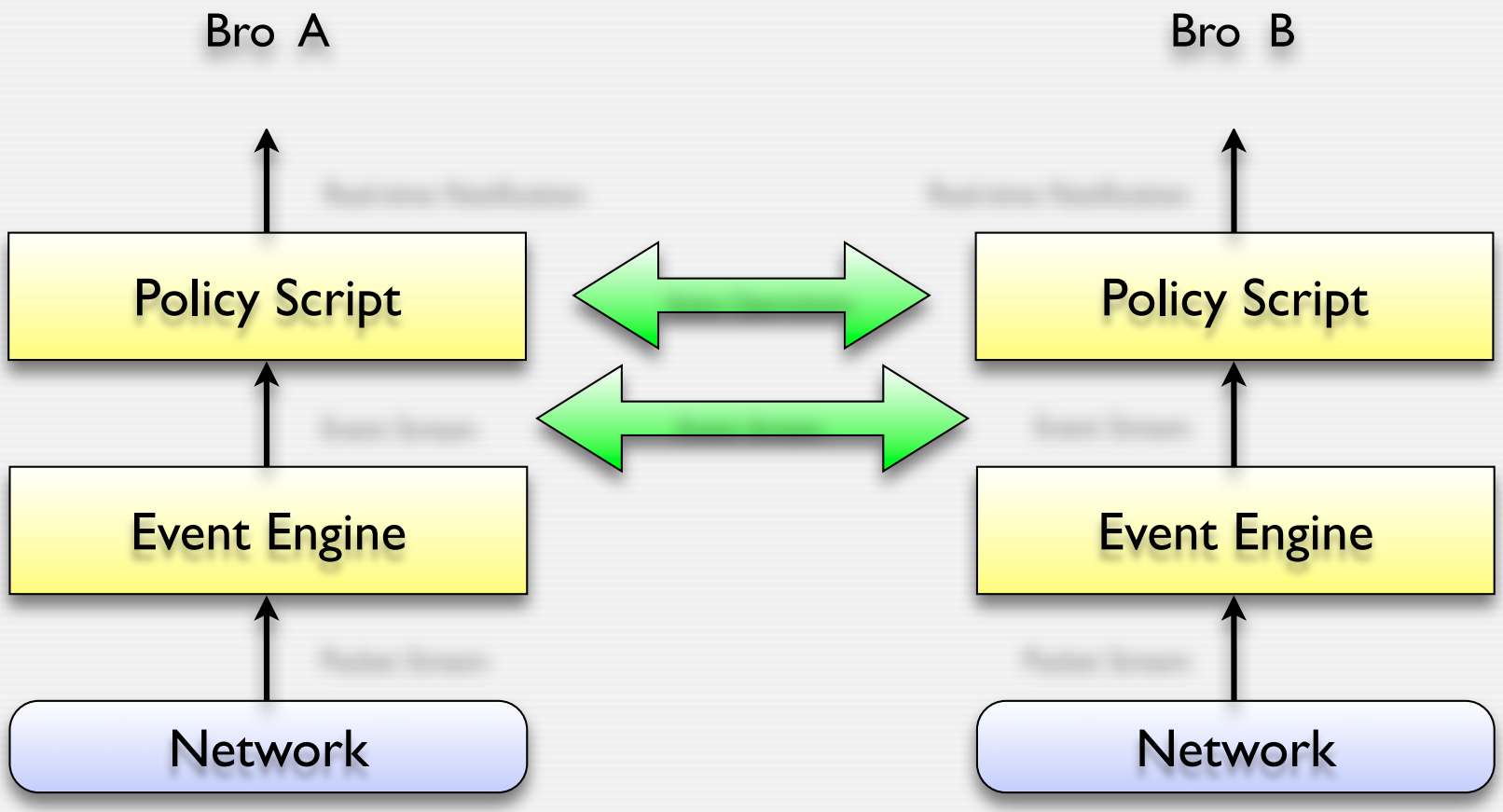
Communication Architecture



Communication Architecture



Communication Architecture



Recent Developments (1)
The Bro Cluster

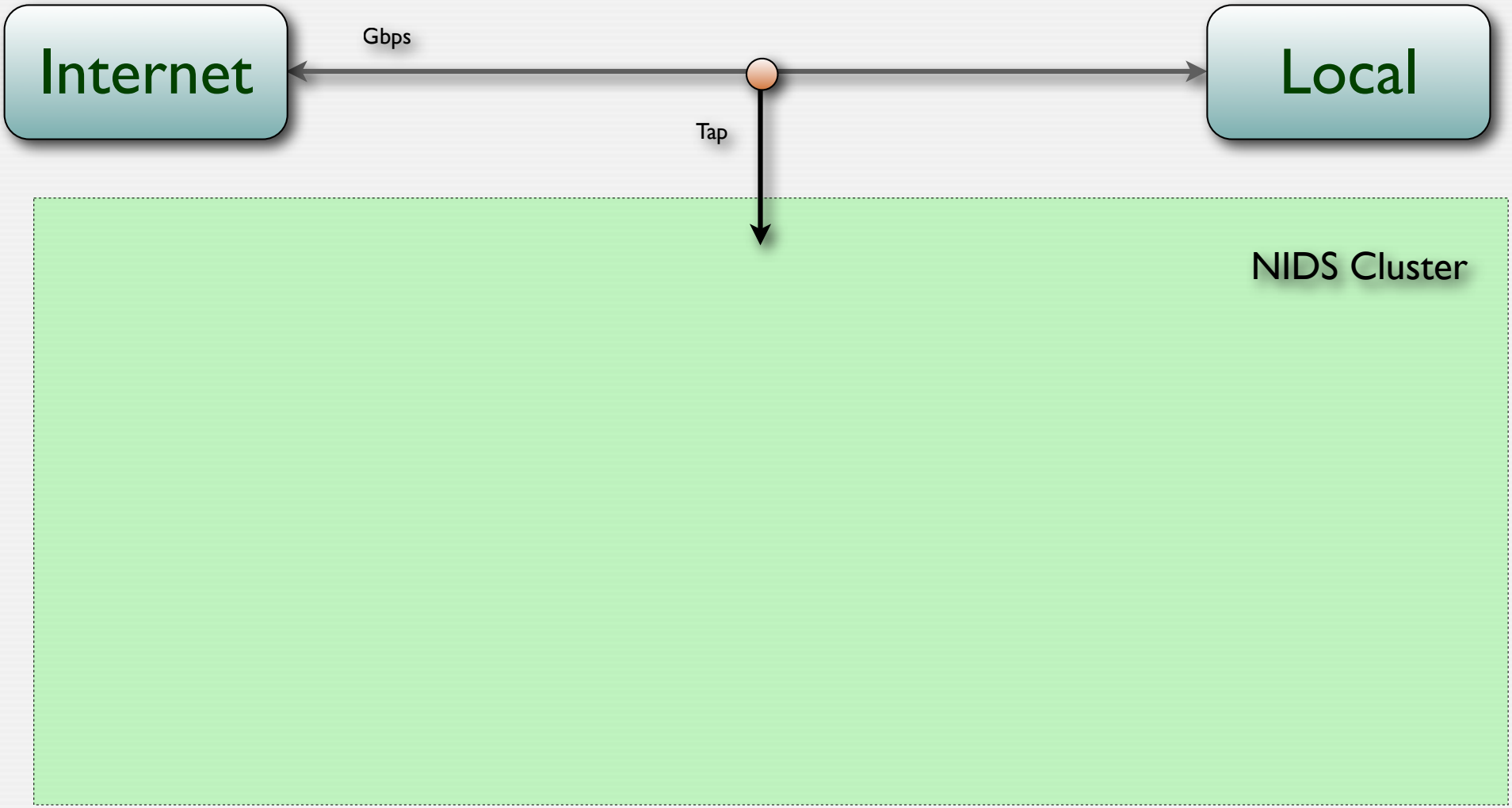
Motivation

- NIDSs have reached their limits on commodity hardware
  - Keep needing to do *more analysis* on *more data* at *higher speeds*
  - Analysis gets richer over time, as attacks get more sophisticated
  - However, single CPU performance is not growing anymore the way it used to
  - Single NIDS instance (Snort, Bro) cannot cope with >=1Gbps links
- Key to overcome current limits is *parallel analysis*
  - Volume is high but composed of many *independent* tasks
  - Need to exploit parallelism to cope with load

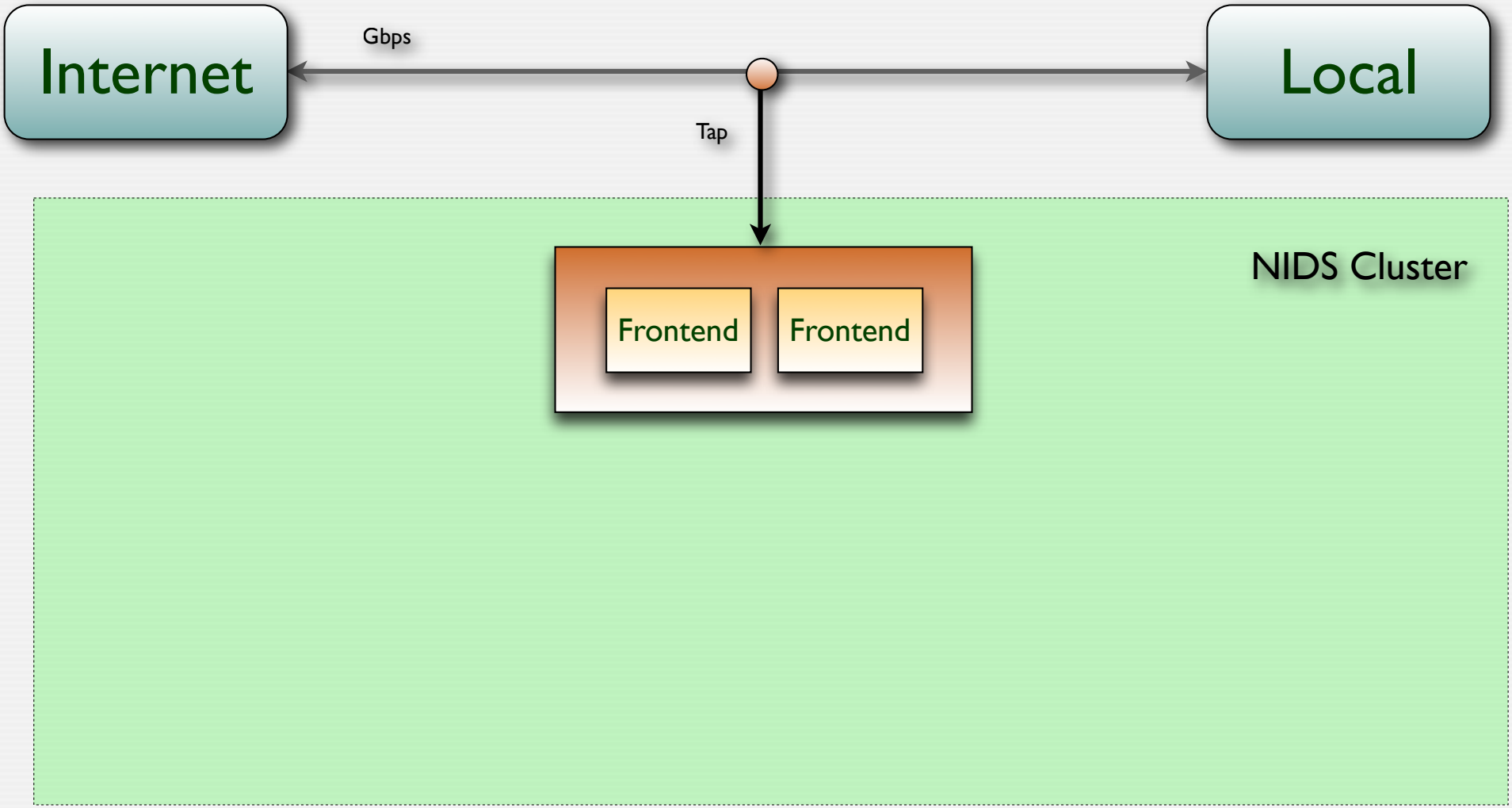
The Bro Cluster

- Load-balancing approach: use many boxes instead of one
- The Bro cluster works *transparently* like a single NIDS
  - Gives same results as single NIDS would if it could analyze all traffic
  - Correlation of low-level analysis
  - No loss in detection accuracy
  - Scalable to large number of nodes
  - Single system for user interface (log aggregation, configuration changes)
- Most NIDS provide support for multi-system setups
- However instances tend to work independently
  - Central manager collects alerts of independent NIDS instances
  - Aggregates results instead of correlating analysis

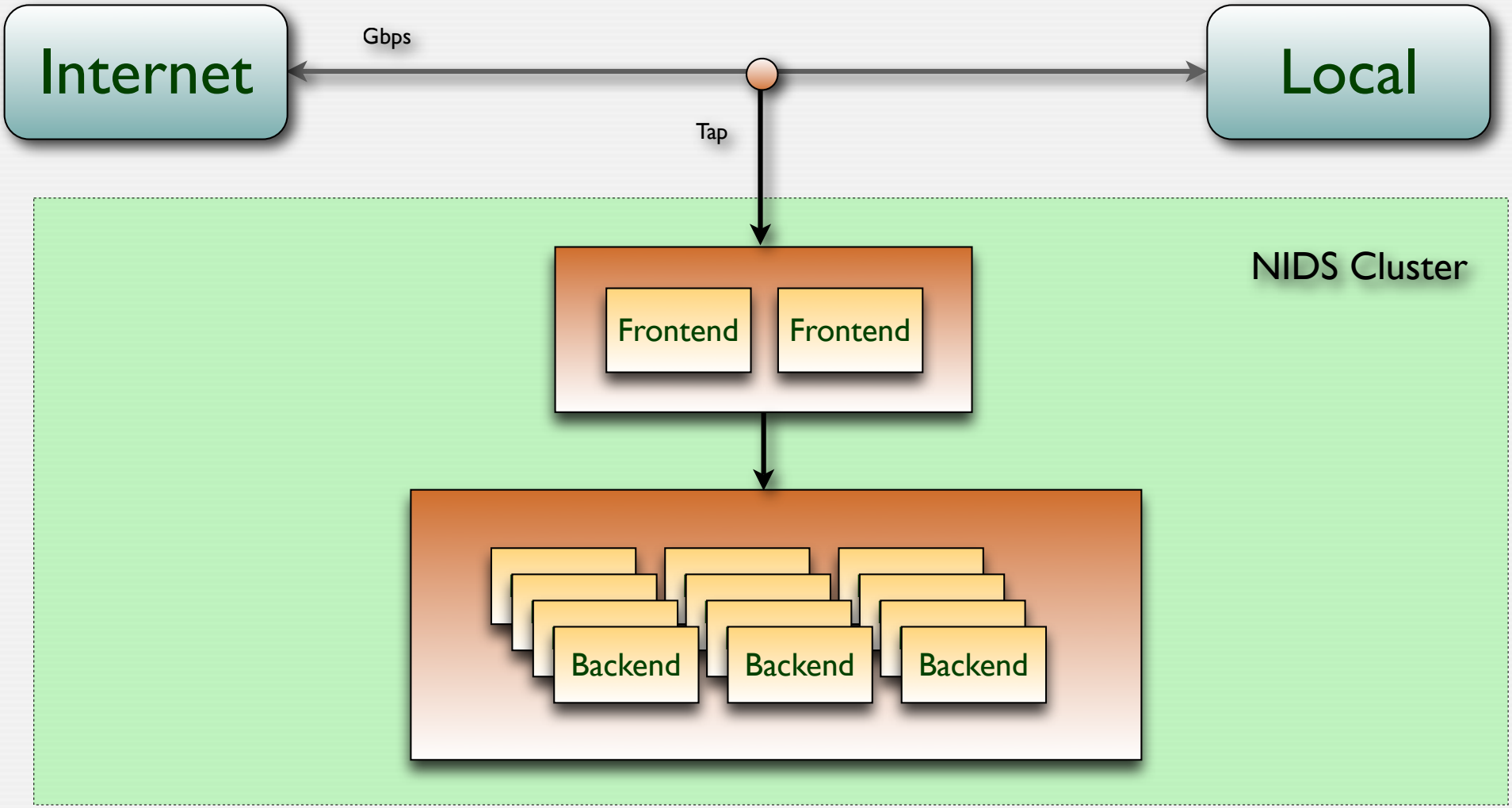
Architecture



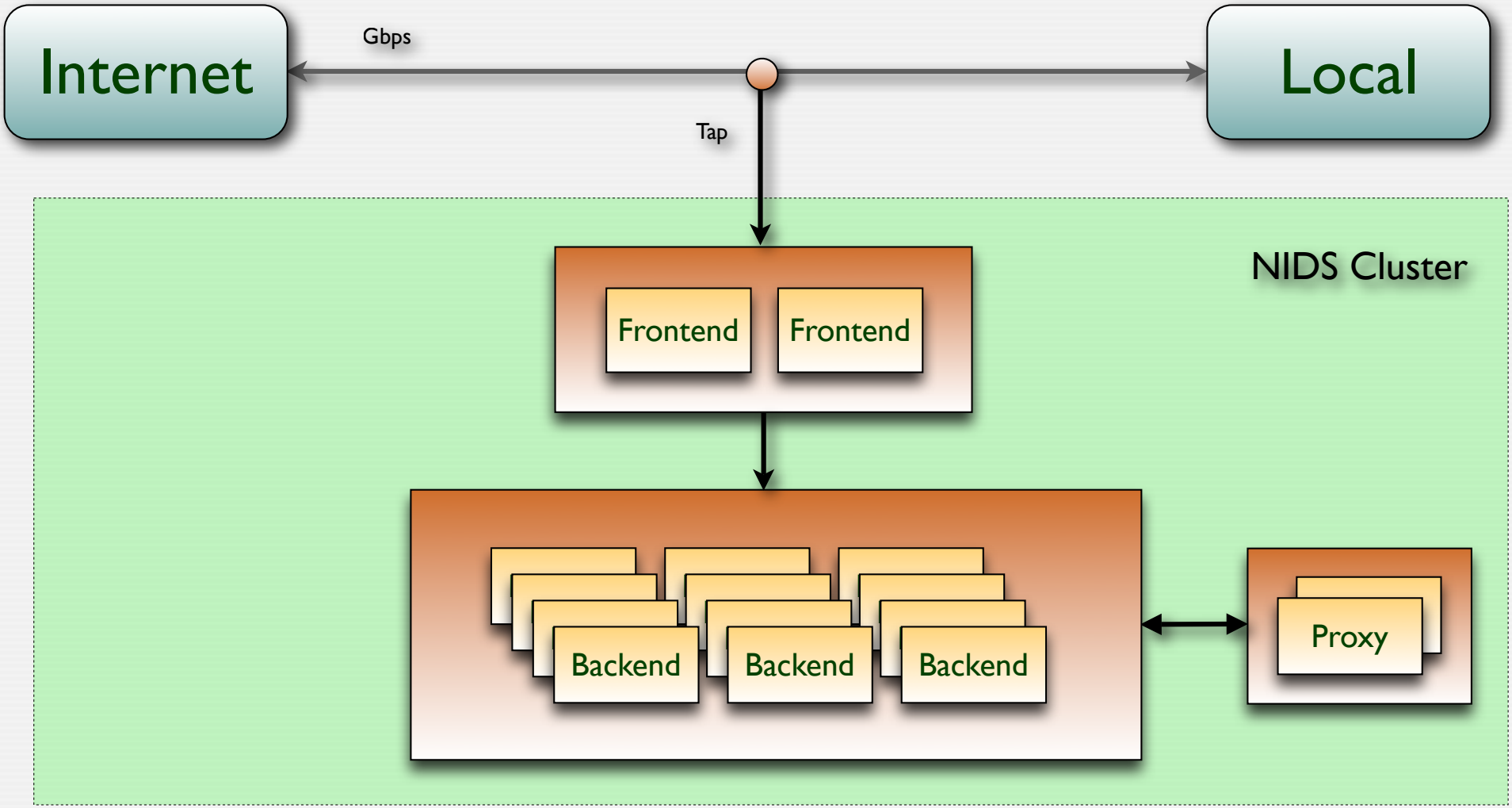
Architecture



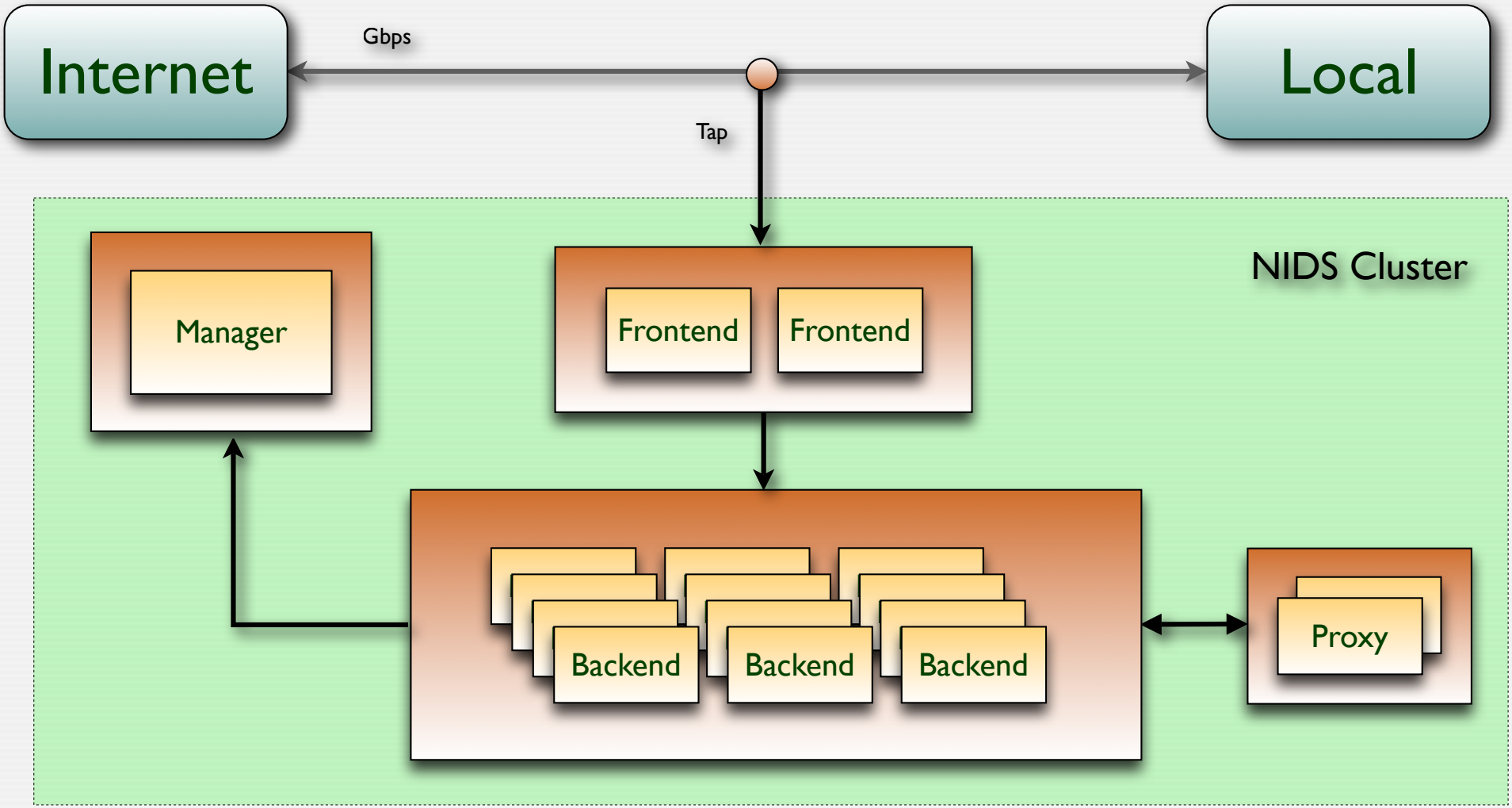
Architecture



Architecture



Architecture



Prototype Setups

- Lawrence Berkeley National Laboratory
  - Monitors 10 Gbps upstream link
  - 1 frontend, 10 backends
- University of California, Berkeley
  - Monitors 2x1Gbps upstream links
  - 2 frontends, 6 backends
- IEEE Supercomputing 2006
  - Monitored conference's 1 Gbps backbone network
  - 10 Gbps *High Speed Bandwidth Challenge* network
- Goal: Replace current operational security monitoring

Frontends

- Distributing traffic to backends by rewriting MACs
  - In software via Click (open-source "modular router")
  - In hardware via Force-10's P10 (prototype in collaboration with F10)
- Slicing the traffic connection-wise
  - Hashing based on either 4-tuple (addrs,ports) or 2-tuple (addrs)
  - MD5 mod n, ADD mod n

Backends

- Running Bro as their analysis engine
- Bro provides extensive communication facilities
  - Independent state framework
  - Sharing of *low-level* state
  - Script-layer variables can be *synchronized*
- Basic approach: pick state to be synchronized
  - A few subtleties needed to be solved
- Central manager
  - Collects output of all instances
  - Raises alerts
  - Provides dynamic reconfiguration facilities

Evaluation & Outlook

- Prototypes are running nicely
  - Are able to perform analysis not possible before
  - E.g., full HTTP analysis & Dynamic Protocol Detection/Analysis
- Evaluation
  - Verified accuracy by comparing against single Bro instance
  - Evaluated performance wrt load-balancing quality, scalability, overhead
- Now in the process of making it production quality
- Management interface
  - Prototype of the *Cluster Shell* as an interactive user interface

The Cluster Shell

homer ~

```
robin@homer:~>cluster

Welcome to BroCluster 0.1

Type "help" for help.

[BroCluster] > status
Name        Type        Status      Host      Pid     Peers  Started
manager     manager     homer       running   3743    9      07 Oct 16:49:53
proxy-1     proxy       homer       running   3781    9      07 Oct 16:50:02
worker-2a   worker      lisa        running   86072   2      07 Oct 16:11:18
worker-2b   worker      lisa        running   86110   2      07 Oct 16:11:19
worker-3a   worker      bart        running   93591   2      07 Oct 16:11:21
worker-3b   worker      bart        running   93629   2      07 Oct 16:11:23
worker-4a   worker      maggie      running   92713   2      07 Oct 16:11:24
worker-4b   worker      maggie      running   92751   2      07 Oct 16:11:26
worker-5a   worker      abraham     running   17416   2      07 Oct 16:11:27
worker-5b   worker      abraham     running   17453   2      07 Oct 16:11:29
[BroCluster] > capstats
Host          mbps        kpps        (10s avg)
192.168.1.5   113.1       20.4
192.168.1.4   186.0       27.1
192.168.1.3   131.4       30.7
192.168.1.6   114.5       21.4
[BroCluster] > analysis
            dns is enabled   -  DNS analysis
            ftp is enabled   -  FTP analysis
      http-body is enabled   -  Analysis of HTTP bodies
    http-header is disabled  -  Analysis of HTTP headers
     http-reply is enabled   -  Server-side HTTP analysis
   http-request is enabled   -  Client-side HTTP analysis
           scan is enabled   -  Scan detection
           smtp is enabled   -  SMTP analysis
[BroCluster] >
```

Recent Developments (2)
Dynamic Protocol Detection

Port-based Protocol Analysis

- Bro has lots of application-layer analyzers
- But which protocol does a connection use?
- Traditionally NIDS rely on ports
  - Port 80? Oh, that's HTTP.
- Obviously deficient in two ways
  - There's non-HTTP traffic on port 80 (firewalls tend to open this port...)
  - There's HTTP on ports other than port 80
- Particularly problematic for security monitoring
  - Want to know if somebody avoids the well-known port

Port-independent Analysis

- Look at the *payload* to see what is, e.g., HTTP
- Analyzers already know how a protocol looks like
  - Leverage existing protocol analyzers
  - Let each analyzer *try to parse* the payload
    - If it succeeds, great!
    - If not, then it's actually another protocol
- Ideal setting: *for every connection, try all analyzers*
- However, performance is prohibitive
  - Can't parse 10000s of connections in parallel with all analyzers

Making it realistic ...

- Bro uses byte patterns to *prefilter* connections
  - An HTTP signature looks for *potential* uses of HTTP
  - Then the HTTP analyzer verifies by trying to parse the payload
  - Signatures can be loose because false positives are inexpensive (no alerts!)
- Other NIDS often ship with protocol signatures
  - These directly generate alerts (imagine reporting all non-80 HTTP conns!)
  - These do not trigger protocol-layer semantic analysis (e.g., extracting URLs)
- In Bro, a match triggers further analysis
- Main internal concept: analyzer trees
  - Each connection is associated with an analyzer tree

Finding Bots

- IRC-based bots are a prevalent problem
  - Infected client machines accept commands from their "master"
  - Often IRC-based but not on port 6667
- Just detecting IRC connections not sufficient
  - Often there is legitimate IRC on ports other than 6667
- DPD allows to analyze all IRC sessions *semantically*
  - Looks for typical patterns in NICK and TOPIC
  - Reports if it finds IRC sessions showing both such NICKs and TOPICs
- Very reliable detection of bots
  - Munich universities use it to actively block internal bots automatically

DPD: Summary & Outlook

- Port-independent protocol analysis
  - Idea is straight-forward, but Bro is the only system which does it
- Bro now has a very generic analyzer framework
  - Allows arbitrary changes to analyzer setup during lifetime of connection
  - Is not restricted to any particular approach for protocol detection
- Main performance impact: need to examine *all* packets
  - Well, that's pretty hard to avoid
- Potential extensions
  - More protocol-detection heuristics (e.g., statistical approaches)
  - Analyze tunnels by pipelining analyzers (e.g., to look inside SSL)
  - Hardware support for pre-filtering (e.g., on-NIC filtering)

Internal Monitoring

Drivers for Internal Monitoring

- Border taps do not see everything
  - E.g., internal traffic, payload of encrypted traffic
- Internal monitoring addresses important threats
  - Credential theft (e.g., SSH keys)
  - Incident detection (e.g., internal scanning)
  - Malicious insiders
  - Auditors(!)
- Internal monitoring at LBNL
  - Intra-subnet
  - Inter-subnet
  - Host instrumentation

Inter-Subnet Monitoring

- Monitoring internal routers & switches
  - That's where traffic traverses subnet boundaries
  - Harder to instrument (packet monitors not really feasible lab-wide)
- Netflow
  - Provides summary records of inter-subnet connections (but no payload)
  - Supported by many brands of routers
  - Central collector system collates received netflow data
  - Analysis with
    - flow-tools & custom scripts provide scan detection capability
    - Prototype of Netflow-enabled Bro
- Monitoring unused subnets (LBNL has a lot!)
  - Routing all traffic to a Bro instance
  - Easily finds braid-dead internal scanners but also misconfigurations
  - Some traffic is routed to a honeyfarm

Intra-Subnet Monitoring

- ARP monitoring under development
- To communicate with other hosts on a subnet, a source host needs the MAC of the destination
  - If you ARP for all addresses on a subnet, you're probably up to no good
- ARPs are broadcast so all hosts see them
  - Having one instrumented host allows to detect such activity
- Installing cheap *Linksys boxes* into subnets
  - Report ARP activity back to a central collector
  - See presentation on Wednesday for more (10:30-11:45AM)

Host-based Monitoring

- Central, mandatory syslogging for *all* Unix hosts
  - Simple configuration, one line is added to /etc/syslog.conf:

    ```
    *.* @syslog.lbl.gov
    ```
  - More than 800 hosts logging, resulting in 1.2GB of data per day
  - Central syslog parser forwards information to Bro & other scripts
  - Provides detection of authentication failures & other signs of attacks
  - Provides archive to search once new criteria become available
  - Working on similar instrumentation for Windows & other
- Instrumented SSH daemons (under development)
  - Back in the old days, it was easy to monitor telnet/rsh sessions
  - SSH prevents NIDS from seeing session content these days
  - Developed instrumented SSHD which forwards content to syslog & Bro

First Detection, then Response

- Detection is great – but systems need to go offline
- Box of tools
  - Insert ACL into border router to cut external connectivity
  - Internal Null-Routing to cut internal connectivity
  - DHCP Deny Boot to get system completely of the network

  - NETS – Network Equipment Tracking System provides contacts & context

Summary & Outlook

LBNL's Monitoring

- LBNL monitors border, internal subnets & hosts
- Monitoring is always work in progress
  - Determine and fill gaps in coverage
  - Increased automation
- Evolve our methods to emerging threats
  - Encrypted communications
  - Covert channels (ICMP, DNS, UDP)
  - Detect unencrypted PII
  - Targeted email attacks
- Exchange of information with partner sites

The Bro NIDS

- Bro is one of the most powerful NIDS available
  - Open-source and runs on commodity hardware
  - While primarily a research system, it is well suited for operational use
  - One of the main components of LBNL's network security monitoring
- Working a various extensions
  - Interactive Cluster Shell for easy installation/operation of a Bro Cluster
  - New analyzers for
    NetFlow, BitTorrent, SIP, XML w/ XQuery support, SSL (rewritten)
  - Time Machine interface (see http://www.net.t-labs.tu-berlin.de/research/tm)
- Turning cluster prototype into production
  - Reimplementing frontends on new platforms
- Long-term plans: multi-core support & data sharing

Any questions ... ?

Robin Sommer
Lawrence Berkeley National Laboratory &
International Computer Science Institute

rsommer@lbl.gov
http://www.icir.org