The Bro Network Intrusion Detection System

Robin Sommer
International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org

Cybersecurity Summit for Large Research Facilities 2009

System Philosophy

- Bro has been developed at ICSI & LBNL 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)

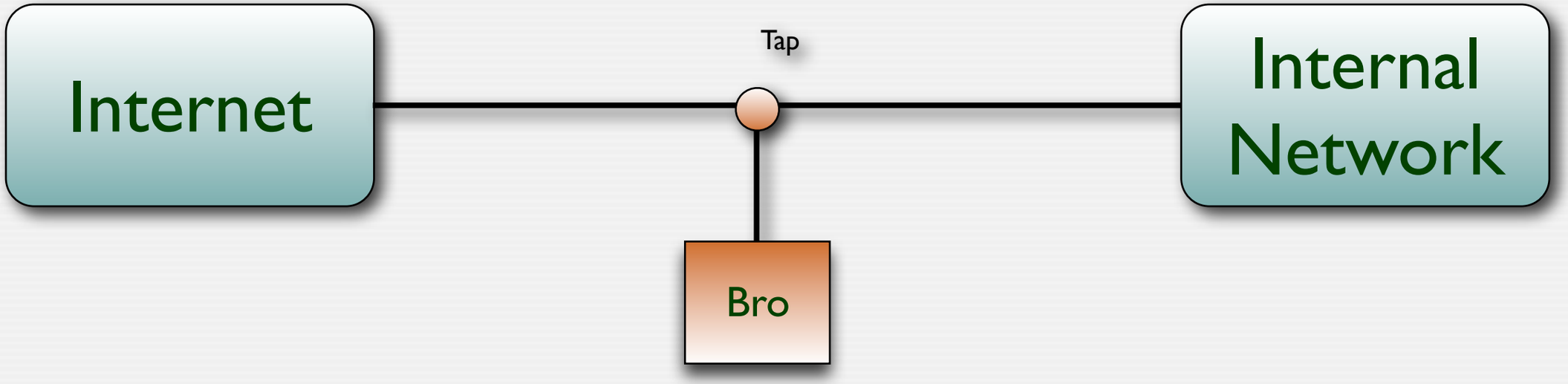
- Operators *program* their policy into the system
  - Not really meaningful to talk about what Bro detects "by default"
- Bro is not restricted to any particular analysis model
  - Most typical is misuse-detection style
- Focus is *not* signature matching
  - Bro is fundamentally different from, e.g., Snort (though it *can* do signatures as well)
- Focus is *not* anomaly detection
  - Though it does support such approaches (and others) in principle
- System thoroughly logs all activity
  - It does not just alert, but keeps an comprehensive archive of activity
  - Logs are invaluable for forensics

Target Environments

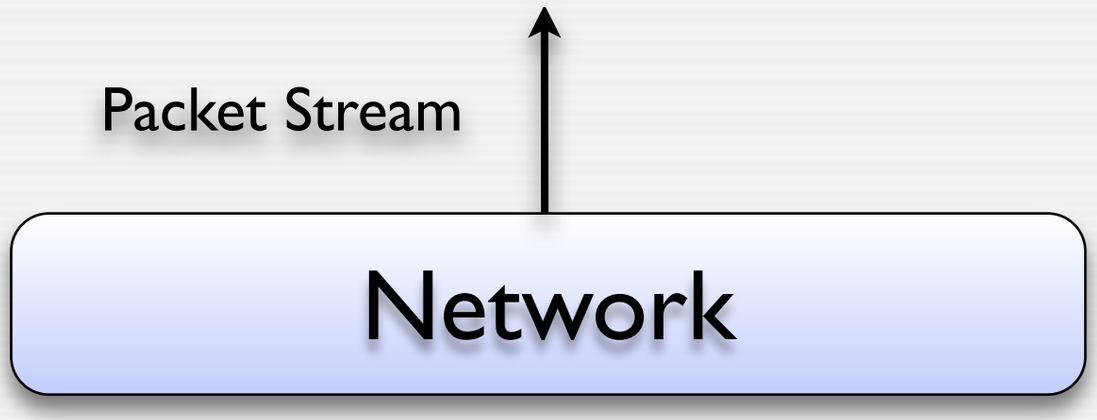
- Bro is specifically well-suited for scientific environments
  - Extremely useful in networks with liberal ("default allow") policies
  - Supports intrusion prevention schemes
  - High-performance on commodity hardware
  - Runs on Unix-based systems (e.g., Linux, FreeBSD, MacOS)
  - Open-source (BSD 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

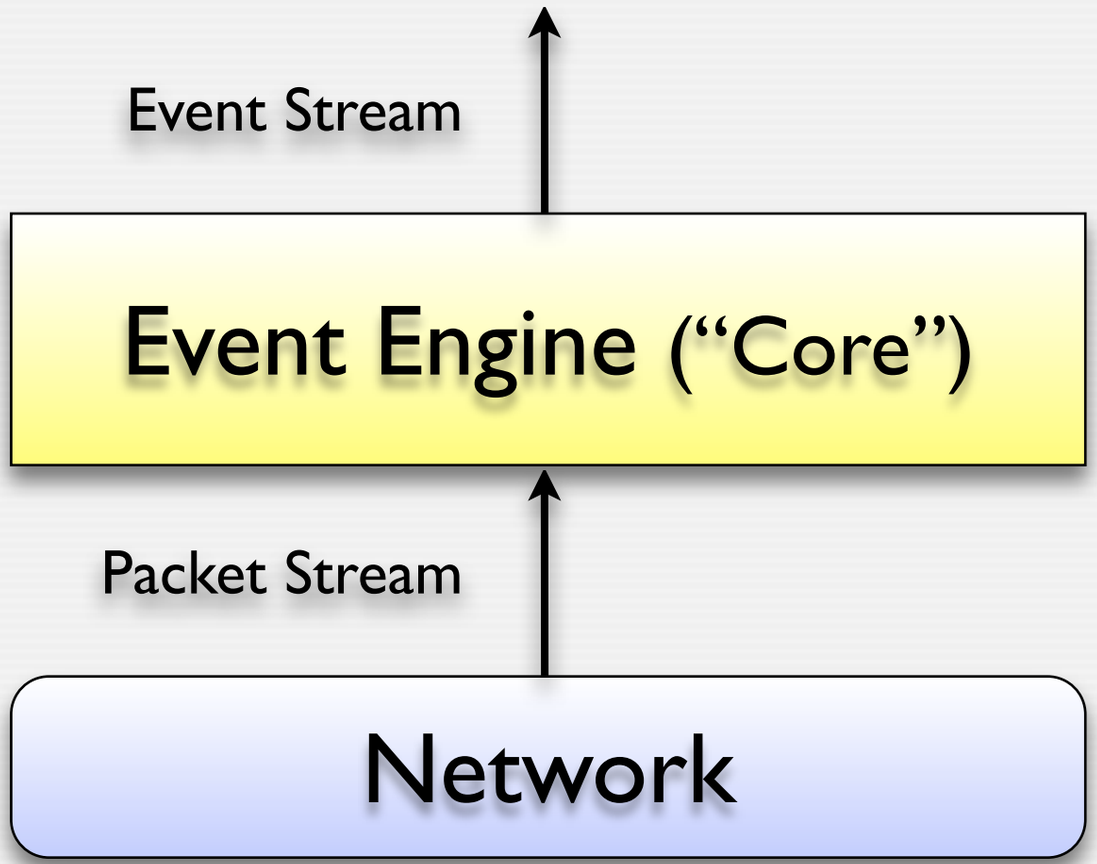
- Bro is typically deployed at a site's upstream link
  - Monitors all external packets coming in or going out
  - Deployment similar to other NIDS
  - By default, purely passive monitoring



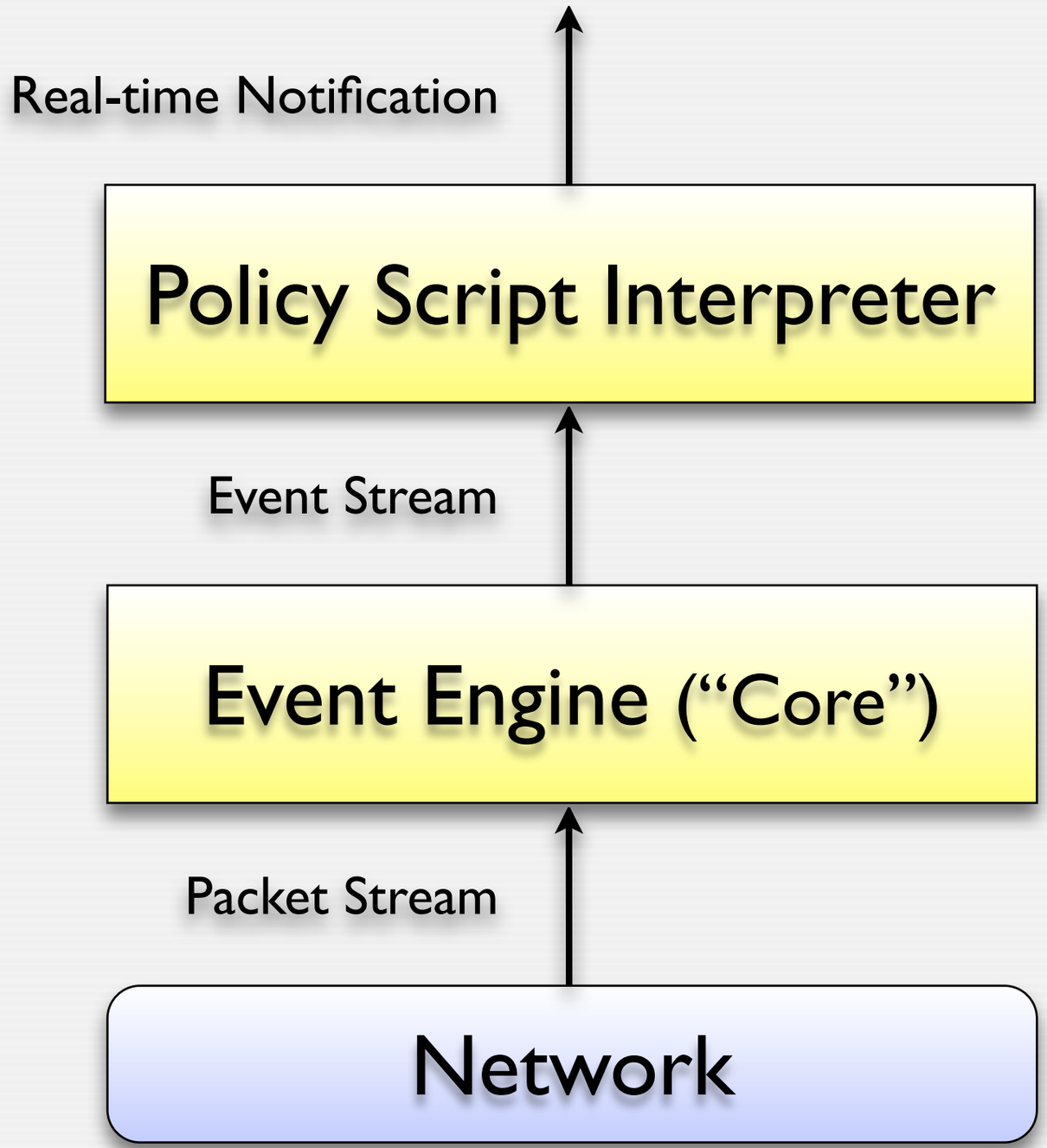
Architecture



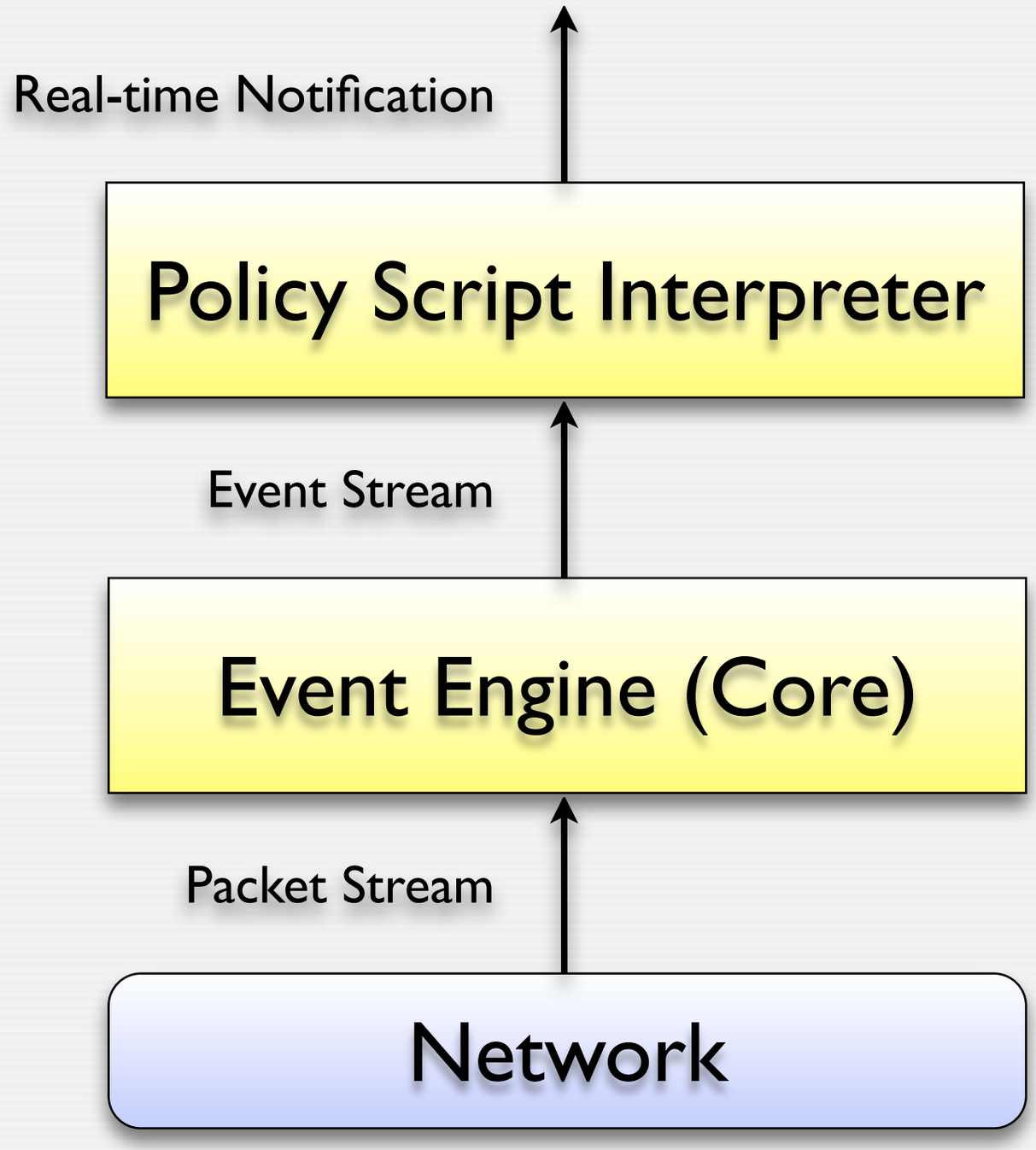
Architecture



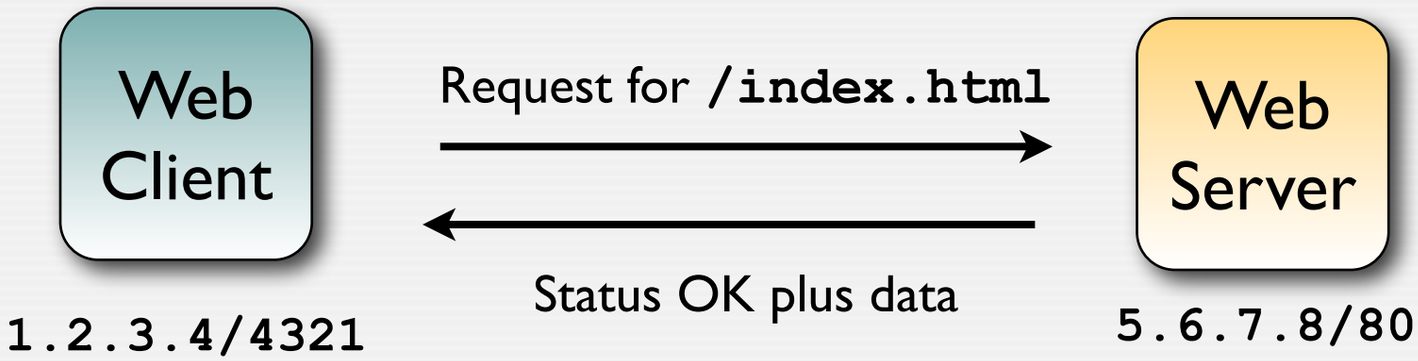
Architecture

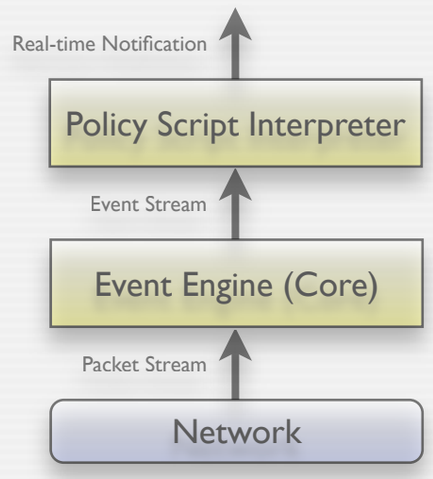


Event Model - Example

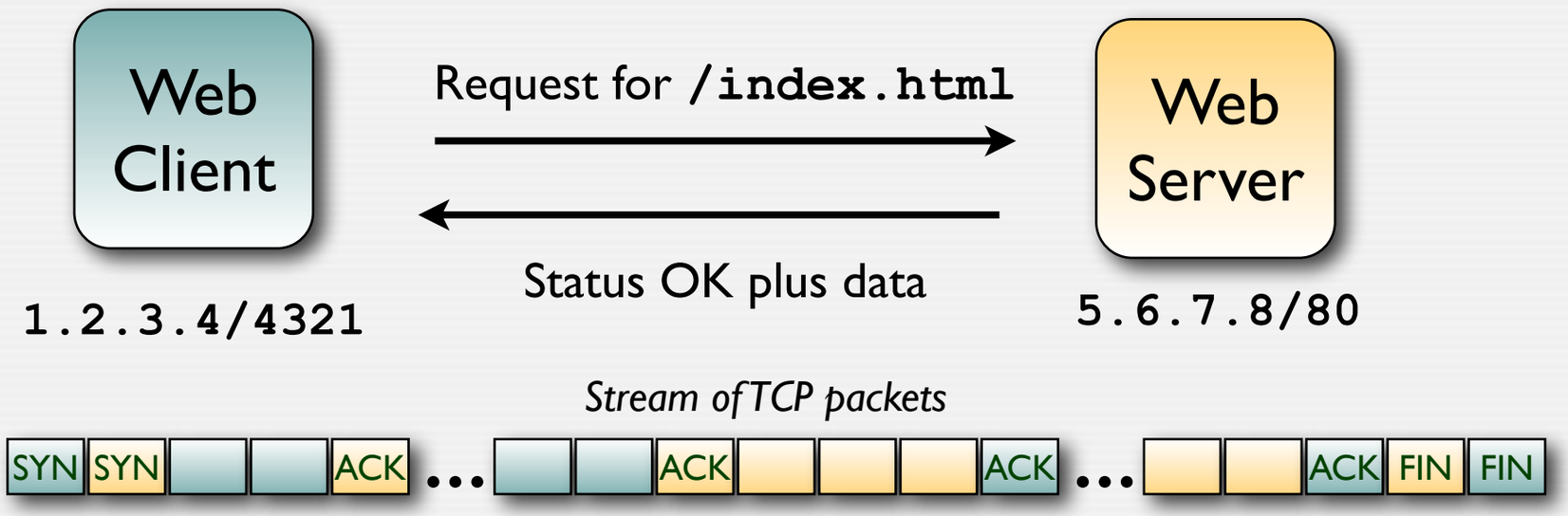


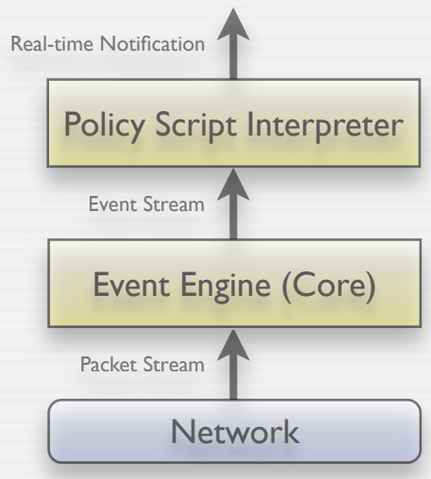
Event Model - Example



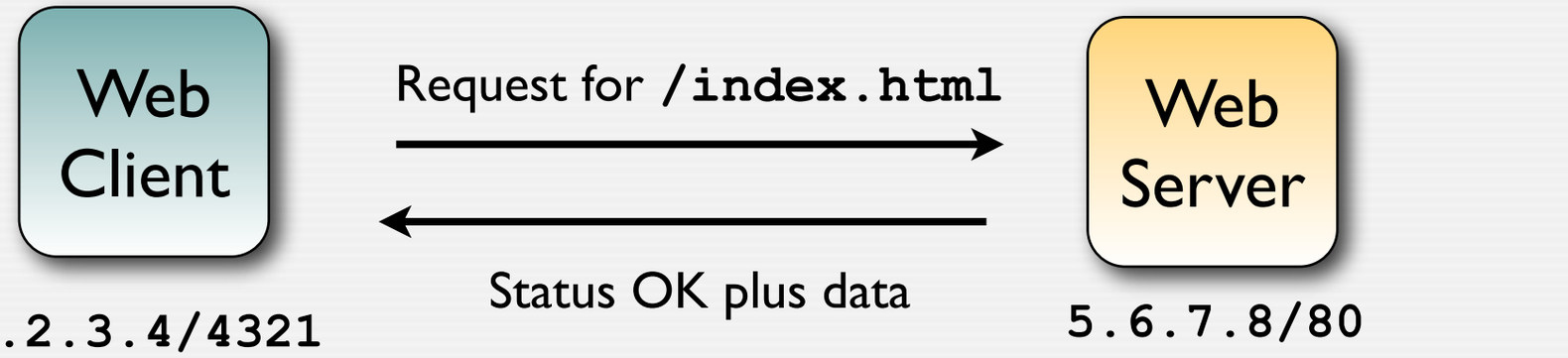


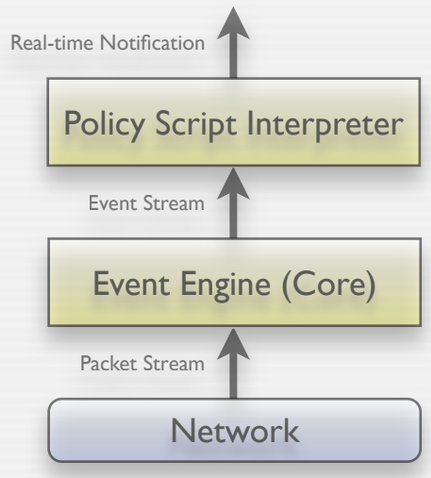
Event Model - Example



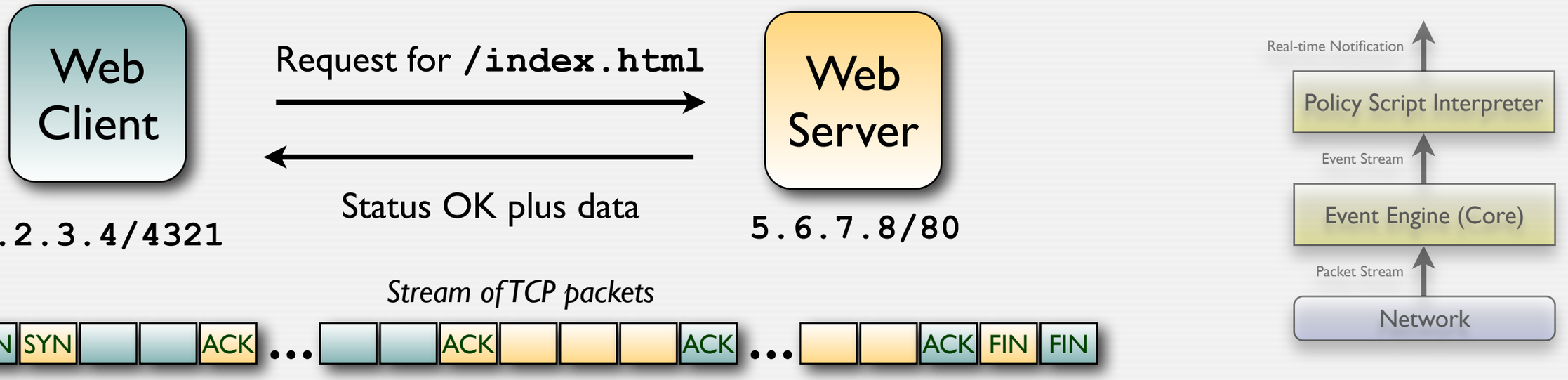


Event Model - Example

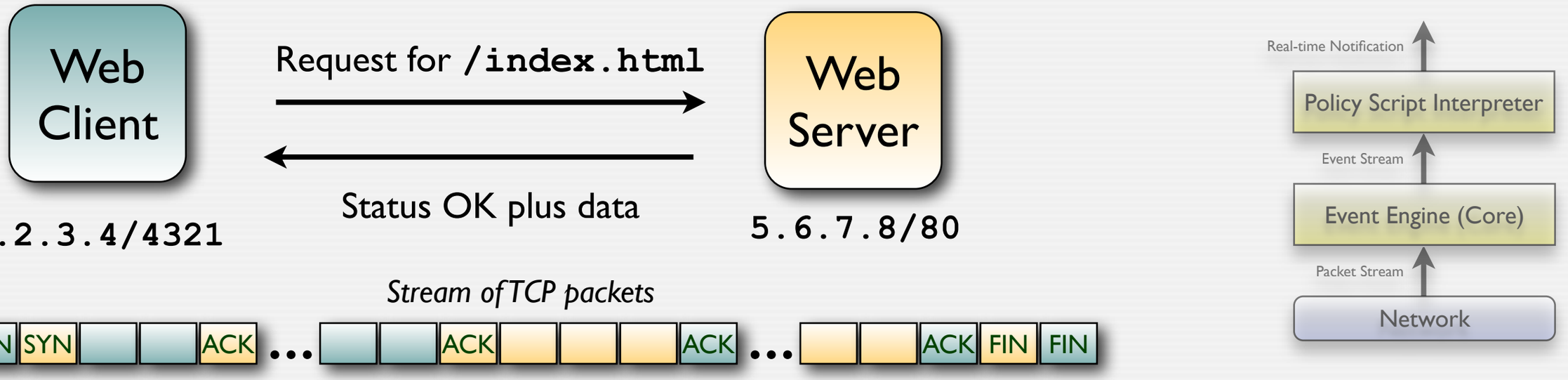




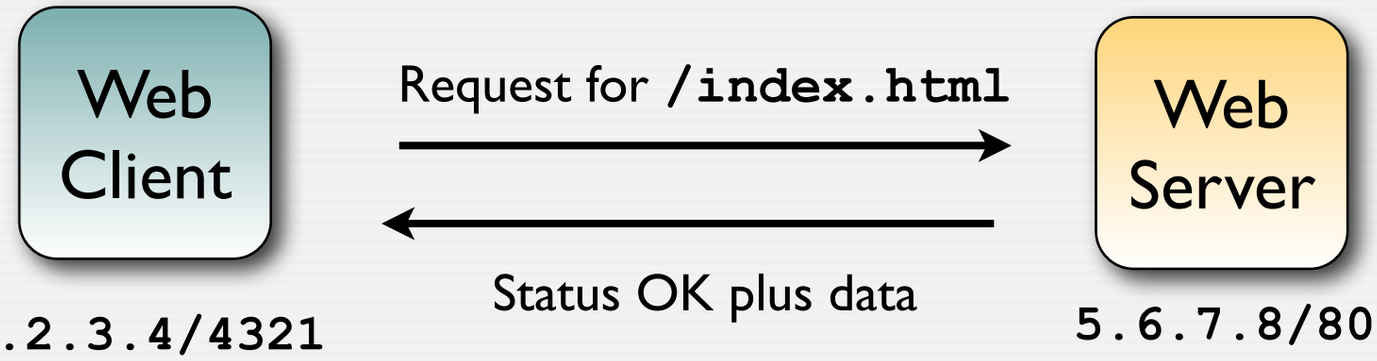
Event Model - Example

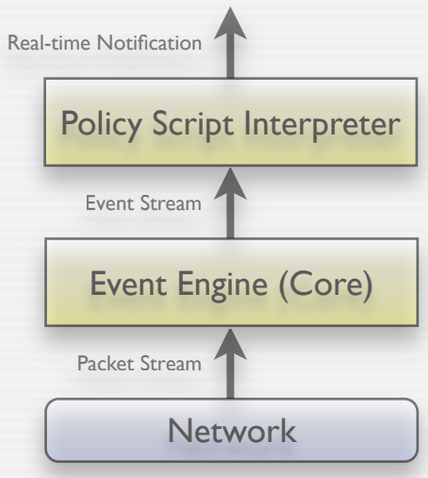


Event Model - Example



Event Model - Example





Stream of TCP packets

Event → `connection established(1.2.3.4/4321→5.6.7.8/80)`

TCP stream reassembly for originator

Event → `http_request(1.2.3.4/4321→5.6.7.8/80, "GET", "/index.html")`

TCP stream reassembly for responder

Event → `http_reply(1.2.3.4/4321→5.6.7.8/80, "OK", data)`

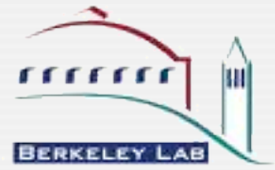
Event → `connection_finished(1.2.3.4/4321, 5.6.7.8/80)`

Event-Engine

- 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)
- Event-engine is written in C++ for performance
  - Performs work *per packet* at line rate
- 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

Expressing Policy with Scripts

- Scripts define policy in terms of these events
- Custom, domain-specific language provides ...
  - Usual script-language types, such as tables and sets
  - Domain-specific types, such as addresses, ports, subnets
  - Extensive state management support, including
    - Timers; automatic expiration; persistence
    - Real-time communication with other Bro instances
- Scripts track network activity
- Scripts take actions
  - Generating alerts via syslog or mail
  - Recording activity to disk
  - Executing external program as a form of response



Script Example: Matching URLs

```
event http_request(c: connection,
                   method: string,
                   path: string)
{
    if ( method == “GET” && path == “/etc/passwd” )
        NOTICE(SensitiveURL, c, path);
}
```

http_request(1.2.3.4/4321→5.6.7.8/80, “GET”, “/index.html”)

Code simplified. See policy/http-request.bro.

Script Example: Tracking SSH Hosts

```
global ssh_hosts: set[addr];

event connection_established(c: connection)
    {
    local responder = c$id$resp_h; # Resp.'s address
    local service = c$id$resp_p;   # Resp'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;
    }
```

Default Policy Scripts

- Large set of default policy scripts
  - Bro ships with 20K+ lines of script code
- Bro's default scripts perform two main tasks
  - Detecting malicious activity (mostly misuse-detection)
  - Logging activity comprehensively without any actual assessment
- In practice, the logs are often most useful
  - Typically we do not know in advance how the next attacks looks like
  - But when an incident occurred, we need to understand what exactly happened
- Typical questions asked
  - *"How did the attacker get in? What damage did he do? Did the guy access other hosts as well? How can we detect similar activity in the future?"*

Example Log: Connection Summaries

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

```
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  Local
```

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

Example Log: HTTP Session

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

A Lot More Features ...

- Signatures à la Snort
- Dynamic protocol detection & analysis
- Remote communication
  - Propagating events, & state synchronization
  - Between Bro instances, and with external programs via *Broccoli* (C/Perl/Python/Ruby)
- Interactive shell for easy configuration & maintenance
  - New in upcoming Bro 1.5 (`broctl`)
- Cluster Operation for 10GE environments
  - Transparent load-balancing solution; moving into production at LBL right now
- Router interface for automatic blocking of IPs
  - Dedicated blocking *appliance*, with Broccoli interface, will be available soon
- "Time Machine" interface
  - A high-performance packet bulk recorder for forensics & retrospective analysis

Upcoming: Bro Hands-On Workshop

- Hands-on Bro training in Berkeley, CA, Oct 13-15
  - Hosted by the Lawrence Berkeley National Lab
  - Primarily targeted at site security personnel
- Informal mix of presentations and hands-on labs
- Bring your laptop!
- More information & registration

  http://www.bro-ids.org/news.html

Thanks for your attention!

Robin Sommer

International Computer Science Institute &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org