U.S. Department of Energy

Office of Science

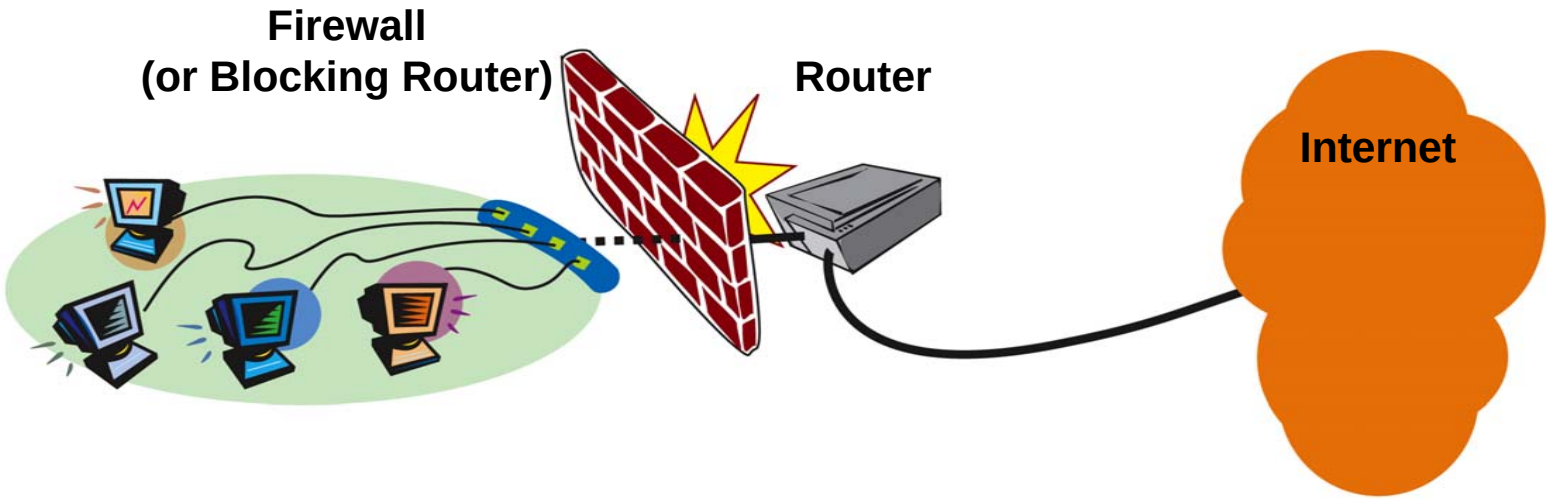
Recent Developments with the Bro Network Intrusion Detection System

Brian L. Tierney

Lawrence Berkeley National Laboratory

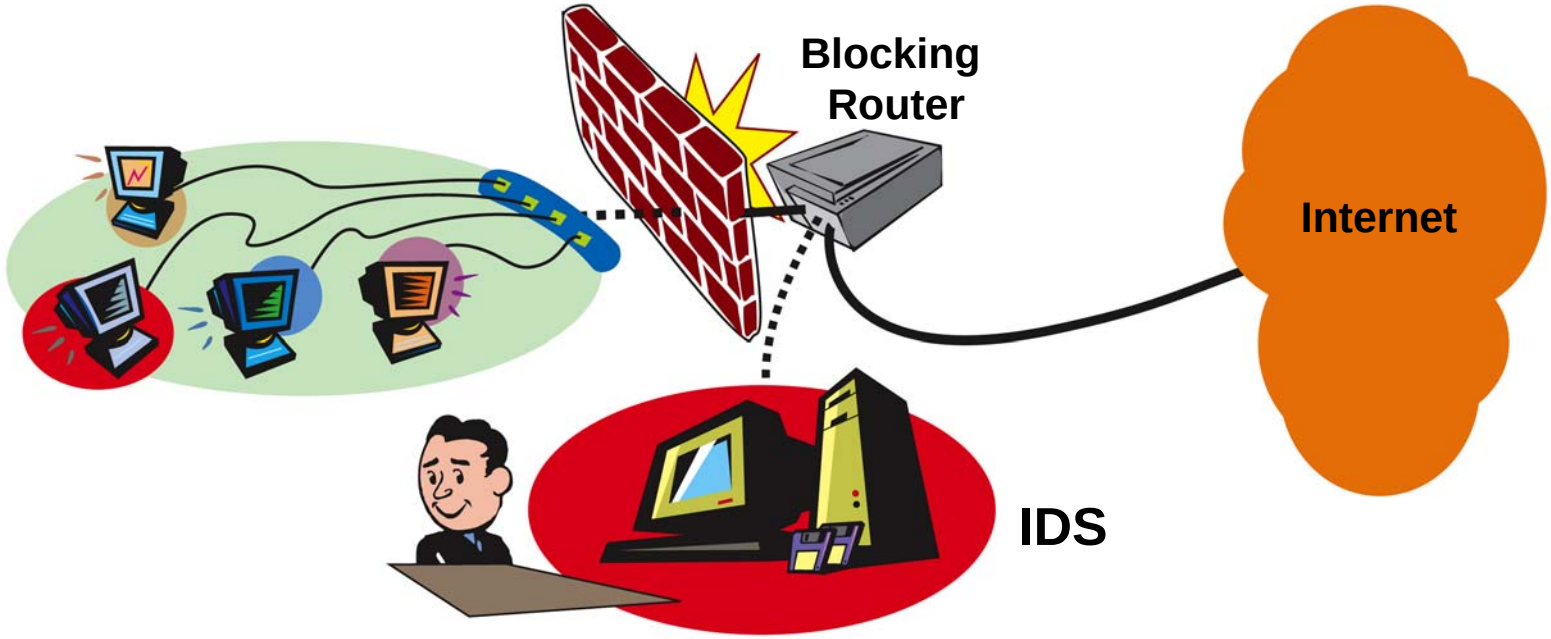
Typical Approach: Firewall with “default deny” policy

- Blocks individual services (ports) inbound and possibly outbound
- Blocks address ranges inbound and possibly outbound



LBNL approach: IDS with Blocking Router

- IDS controls the blocking router
- IDS blocks dynamically when an intrusion attempt is detected or alerts upon suspicious activity
- Router blocks statically like a firewall
- Provides a form of “Intrusion Prevention”



Bro's Design Targets Open Research Environments

- Bro is Designed for the Flexible Research Environments
  - The Grid
  - Widespread collaborations
  - Shared National User Facilities

- 10 years have been invested in optimizing Bro for Open Environments
  - In production use at a number of sites:
    - LBL, NERSC, ESnet, NCSA, UC Davis: Primary IDS
    - Sandia, UCB, TUM, OCCS(ORNL), NOAA: Secondary IDS
  - Runs on low-cost commodity hardware
  - Provides real-time detection and response
  - Ability to monitor traffic in a very high performance environment
  - Ability to write custom policy analyzers

Bro's Use at LBL

- Operational 24×7 since 1996
- Monitors traffic for suspicious behavior or policy violations: incoming/outgoing/internal
- In conjunction with blocking routers, Bro *acts* as a dynamic and intelligent firewall
  - Blocks access from offending IP addresses
  - Blocks known hostile activity
  - Terminates connections and/or sends alarms
  - Locates site policy violations (e.g.: Kazaa and gnutella)

Styles of intrusion detection — Signature-based:

- Core idea: look for specific, known attacks.
- Example (from Snort):
  - `alert tcp $EXTERNAL_NET any -> $HOME_NET 139 flow:to_server,established`
  - `content:"|eb2f 5feb 4a5e 89fb 893e 89f2|"`
  - `msg:"EXPLOIT x86 linux samba overflow"`
  - `reference:bugtraq,1816`
  - `reference:cve,CVE-1999-0811`
  - `classtype:attempted-admin`

- Most commercial system (e.g.: ISS RealSecure) are Signature-based
  - Signatures can be at different semantic layers, e.g.: IP/TCP header fields; packet payload; URLs.
- Pros
  - good attack libraries, easy to understand results.
- Cons:
  - unable to detect new attacks, or even just variants.

Styles of intrusion detection — Anomaly-detection

- Core idea: *attacks are peculiar*.
  - Approach: build/infer a profile of “normal” use, flag deviations.
  - Example: “user joe only logs in from host A, usually at night.”

- Pros:
  - potentially detects wide range of attacks, including previously unknown types of attacks.
- Cons:
  - potentially misses wide range of attacks, including known.
  - can potentially be “trained” to accept attacks as normal
  - often large number of false positives

Styles of detection — Activity-based

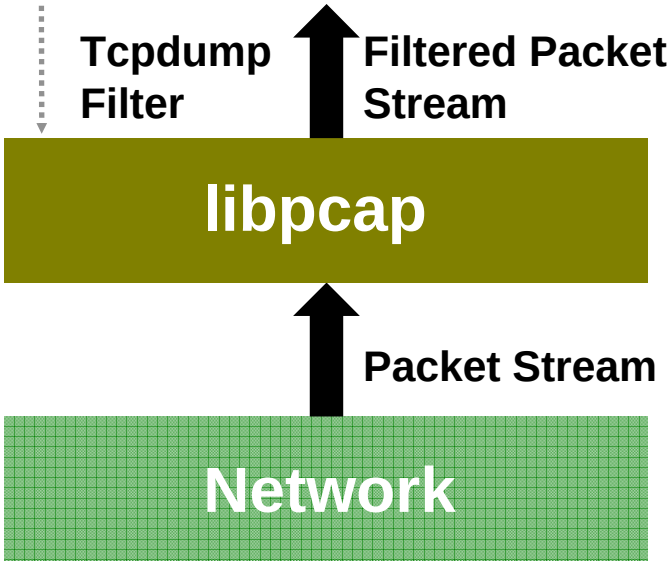
- Core idea: inspect traffic and construct “events”, look for patterns of activity that deviate from a site’s policy.
  - Example: “user joe is only allowed to log in from host A.”
  - Note: this is the primary approach used by Bro.

- Pros
  - potentially detects wide range of attacks, including novel.
  - framework can accommodate signatures, anomalies.
- Cons
  - policies/specifications require significant development & maintenance. Harder to construct attack libraries.

How Bro Works

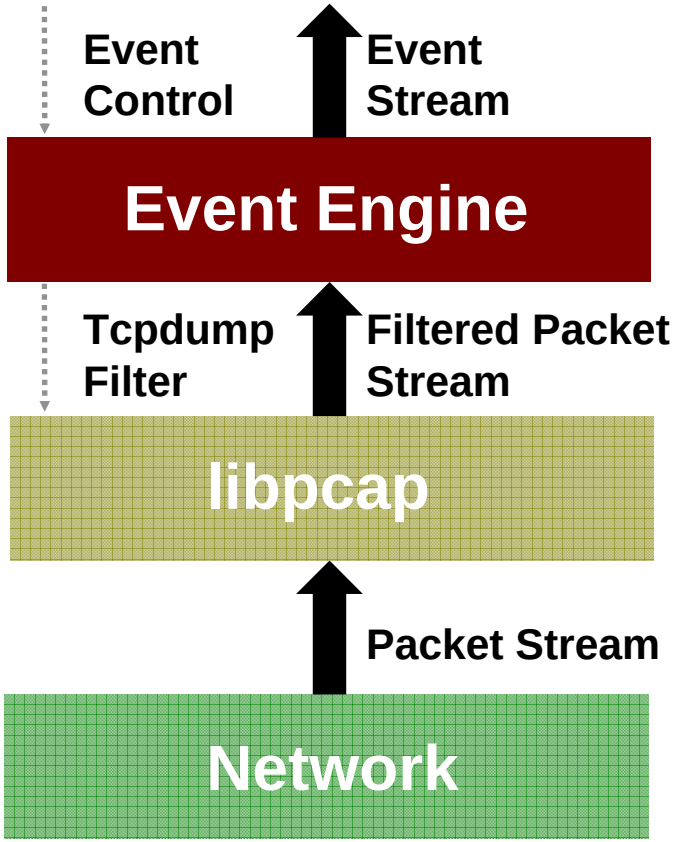
- Taps GigEther fiber link passively, sends up a copy of all network traffic.

How Bro Works



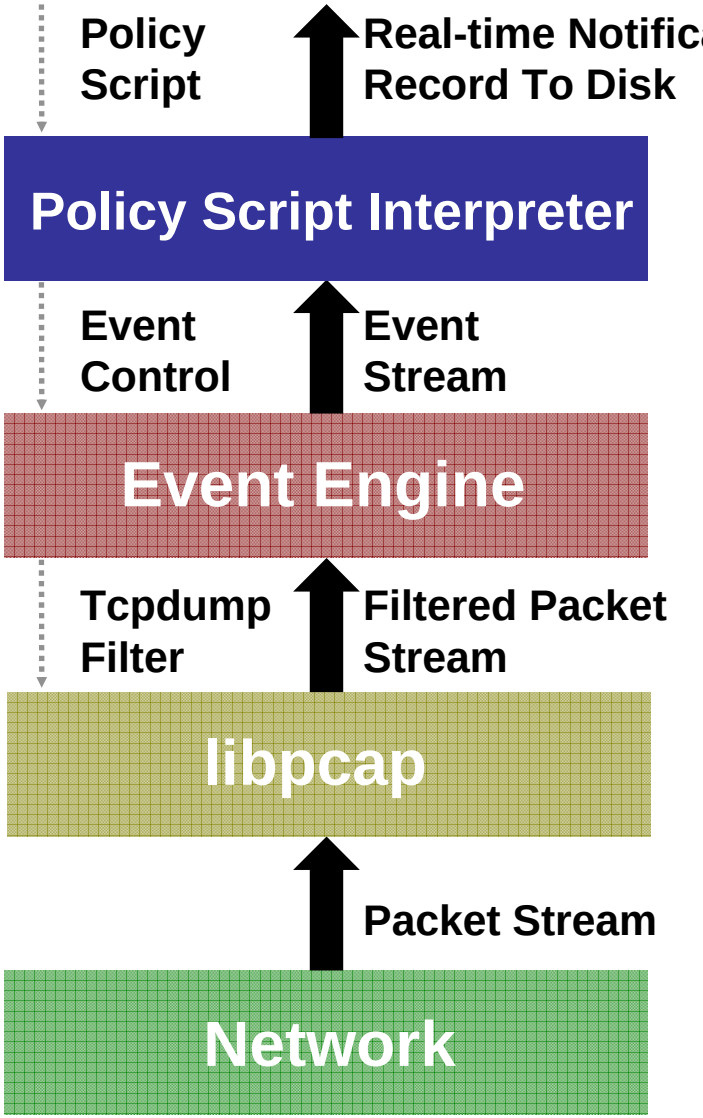
- Kernel filters down high-volume stream via standard *libpcap* packet capture library.

How Bro Works



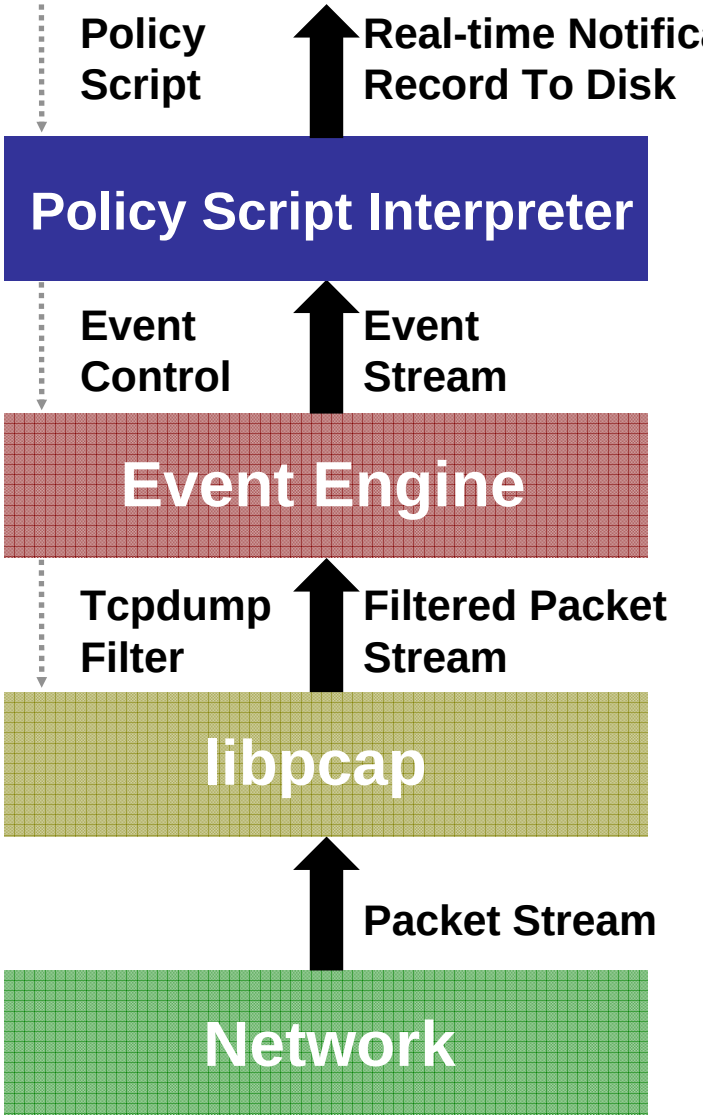
- "Event engine" distills filtered stream into high-level, *policy-neutral* events reflecting underlying network activity
  - E.g. Connection-level:
    - connection attempt
    - connection finished
  - E.g. Application-level:
    - ftp request
    - http_reply
  - E.g. Activity-level:
    - login success

How Bro Works



- "Policy script" processes event stream, incorporates:
  - Context from past events
  - Site's particular policies

How Bro Works



- "Policy script" processes event stream, incorporates:
  - Context from past events
  - Site's particular policies
- … and *takes action*:
  - Records to disk
  - Generates alerts via *syslog*, paging, etc.
  - Executes programs as a form of response

Bro Protocol Analyzers

- Bro includes the following protocol analyzers
  - full analysis:
    - HTTP, FTP, telnet, rlogin, rsh, RPC, DCE/RPC, DNS, Windows Domain Service, SMTP, IRC, POP3, NTP, ARP, ICMP, Finger, Ident
  - partial analysis:
    - NFS, SMB, NCP, SSH, SSL, TFTP, Gnutella
  - in progress:
    - AIM, BGP, DHCP, Windows RPC, SMB, NetBIOS, NCP

Sample Bro Policy

- Using the Bro language, sites can write custom policy scripts to generate alarms on any policy violation.
- For example, if a site only allows external http and mail to a small, controlled lists of hosts, they could do this:

```
const web_servers = { www.lbl.gov, www.bro-ids.org, };
const mail_servers = { smtp.lbl.gov, smtp2.lbl.gov, };

redef allow_services_to: set[addr, port] += {
        [mail_servers, smtp],
        [web_servers, http],
};
```

- Bro can then generate an *Alarm* or even terminate the connection for policy violations:

```
if ( service !in allow_services)
   NOTICE([$note=SensitiveConnection, $conn=c,]);
if ( inbound && service in terminate_successful_inbound_service )
     terminate_connection(c);
```

Recent Advances in Bro

Bro Communication

- New **Bro co**mmunication **li**brary (Broccoli)
  - Multiple Bro's can now communicate and exchange "events"
  - Can easily synchronize bro state between running instances of Bro
    - Just add "&synchronized" to the table declaration

      ```
      global scan_hosts: table[addr] of count &synchronized;
      ```

    - This can be used to maintain a table of known hostile hosts between multiple instances of Bro
  - Using Broccoli to send syslog events to Bro
    - LBNL runs a central syslog collector for entire lab
      - Subset of these logs are send to Bro for analysis (ssh, su, sudo, etc.)
    - Bro policy is being used to analyze syslog logs
      - E.g.: multiple ssh login failures, offsite root logins, etc.
  - Modified sshd that sends data to Bro directly
    - Can use Bro's "login" analyzer to look for suspicious commands
      - E.g: 'unset history'

Dynamic Application Detection

- Current NIDS system require you to specify which protocol analyzer to use for a given port.
  - I.e: port 25 = SMTP; port 80 = HTTP, port 6666 = IRC, etc.
- NIDS's only look at traffic on ports they know how to analyze
- New version of Bro supports dynamic port selection
  - Uses simple protocol-specific signatures to try to guess what protocol is being seen
    - Enhanced version of "Layer 7 packet classifier"
      - http://L7-filter.sourceforge.net
  - Sample use at LBL:
    - Look for http proxies
    - Look for FTP and SMTP on non-standard ports
    - Looks for IRC "botnets"
    - payload inspection of FTP data transfers
  - Note: dynamic application detection takes more CPU and IO because it looks at all traffic,
    - may need a dedicated Bro host for this at medium/large sites

Dynamic HTTP Analyzer

- HTTP analyzer can distinguish the various protocols that use HTTP as their transport protocol by looking for their characteristics
  - Includes patterns for detecting Kazaa, Gnutella, BitTorrent, Squid, and SOAP applications running over HTTP
  - The HTTP analyzer extracts the “Server” header from the HTTP responses

- Examples:
  - ProtocolFound 66.249.65.49/62669 > 131.243.224.47/1400 FileMakerPro (via HTTP) on port 1400/tcp
  - ProtocolFound 66.249.66.177/47957 > 131.243.2.93/8881 Apache (via HTTP) on port 8881/tcp
  - ProtocolFound 198.129.90.45/1160 > 128.3.72.29/7777 Oracle (via HTTP) on port 7777/tcp
  - ProtocolFound 211.37.103.215/1278 > 131.243.129.75/554 RealServer (via HTTP) on port 554/tcp

Payload inspection of FTP Data Transfers

- Attackers often install FTP-servers on non-standard ports on compromised machines
- Analysis of FTP data connections is impossible with traditional NIDSs
  - FTP uses arbitrary port combinations for data connections.
- The Bro file analyzer receives the file's full content and can utilize any file-based intrusion detection scheme.
  - includes file-type identification to Bro using `libmagic`
    - can identify a large number of file-types
    - E.g.: Bro is now able to categorize a data file as being of MIME type `video/x-msvideo` (an AVI movie)

Detecting IRC-based Botnets

- Attackers systematically install trojans *bots* on compromised machines
  - large *botnets* provide remote command execution on vulnerable systems across the world.
- A botnet is usually controlled by a *master* that communicates with the bots by sending commands.
  - E.G.: flood a victim, send spam, or sniff confidential information such as passwords
- Botnets are one of the largest threats in today's Internet

- The IRC protocol is popular for bot communication
  - It is extremely difficult for a traditional NIDS to reliably detect members of IRC-based botnets.
  - Often, the bots never connect to a standard IRC server port.

Botnet Detector

- The detector sits on top of the IRC analyzer and is therefore able to perform protocol-aware analysis of *all* detected IRC sessions.
- To identify a bot connection, it uses three heuristics.
  - First, it checks if the client's nickname matches a (customizable) set of patterns known to be used by botnets
  - Second, it examines the channel topics if it includes a known typical botnet commands
  - Third, clients that establish an IRC connection to an already identified bot-server are also considered to be bots.
    - This is very powerful as it leverages the state that the detector accumulates over time, and does not depend on any particular payload pattern.
- Successfully located several botnets at Technical University Munich and UC Berkeley

Current Work

- Using Bro to detect *insider attacks*
- Deploy Bro on every subnet
  - Using $60 Linksys routers running Linux for 100BT
  - Using $300-$400 PCs for 1000BT networks

- Easier to define more restrictive per-subnet policy than per-site policy

Deploying Bro to Monitor Grid Services

- Grid Services could tell Bro (via Broccoli) which hosts/ports it is using
  - Could general alerts for any traffic on unexpected ports
- Use Bro to verify firewall configuration
  - Including newly proposed dynamic firewall configuration
- Auditing Tool
  - Keep track of connections and bytes transferred between grid sites
- Forensics Tool

For more Information

- Bro is Open Source (FreeBSD-style license)
  - Download from: http://www.bro-ids.org/
  - Dynamic Application Detection and Time Machine will be in the next release (by summer)

- Questions: email bro@bro-ids.org

Extra Slides