Bro at 10 Gps: Current Testing and Plans

Brian L. Tierney

Lawrence Berkeley National Laboratory
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)
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 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_to)
      NOTICE([$note=SensitiveConnection, $conn=c,]);
  if ( inbound && service in terminate_successful_inbound_service )
      terminate_connection(c);
  ```
Recent Advances in Bro
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”
  - 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,
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
The IRC protocol is popular for bot communication, but
   – It is difficult for a traditional NIDS to reliably detect members of IRC-based botnets.
   – Often the bots don’t use standard IRC server ports.
The Botnet detector sits on top of the Bro 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.
Successfully located several botnets at Technical University Munich and UC Berkeley
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)
Bro at 10GBPS
There are a number of approaches to Bro at 10 GigE

- **Use a 10 Gig NIC**
  - Pros: will just work; no special configuration needed
  - Cons: cost, even with very high end host may run out of CPU cycles for analysis
  - Performance: Current testing shows that 10 Gig Nics can only capture about 1 Gbps of traffic (more testing in progress now)

- **Use Filter Based Port Mirroring or VACLs.**
  - Forward only interesting traffic to a Bro host
  - Pros: works well with standard Bro configuration
  - Cons: requires Juniper routers or CISCO 65xx router and spare 1 Gig port in the router

- **Use Custom Hardware**
  - ICIR shunting project: see next slide
  - Other custom hardware
    - Force10 (former Metanetworks) 10G nic provides on NIC BPF filtering
Shunting Project

• Based on the following concept:
  – For large flows, the beginning of the connection contains nearly all the information of interest from a security analysis perspective
• Shunting uses a simple in-line hardware element that maintains several large state tables indexed by packet header fields
  – Based on table value, every packet is either:
    • forwarded, dropped, or diverted through the IPS
  – Low cost FPGA-based
    • Xilinx Virtex 2 Pro FPGA coupled to 2 independent 2MB SRAM banks and a control FPGA
Shunting Approach

External Network

Control & Analysis Traffic

Filter

GigE Interface

Router / FPGA

Internal Network

Bro
Force10 P10 Box: Early Results

• Configured a Force10 P10 to capture only the 1st 10 packets of a flow (each direction)
• Results:
  – reduced the number of packets passed to the OS by around 95%
  – Reduced the number of interrupts tremendously
  – Reduced the CPU needed by Bro by 4x
Future Plans: Bro Cluster

- Front end node uses Force10 hardware to rewrite packets, forwarding them to cluster nodes for analysis
- Classify packets and forward to specific ethernet address
  - Could use multicast address to allow for transparent node redundancy
- Scalable in several ways:
  - As traffic increases, just add more nodes
  - For more detailed Bro analysis, just add more nodes
  - Should scale to 100 Gbps networks and beyond
  - Could run Bro, Snort, and other future tools all on the same cluster
- New version of Bro has support for synchronized state across multiple Bro systems
  - Have been testing this using packet traces
  - Will soon start testing at 1 Gbps using live traffic
For more Information

• Bro is Open Source (FreeBSD-style license)
  – Download from: http://www.bro-ids.org/
  – Dynamic Application Detection will be in the next release of Bro (September?)

• Questions: email bro@bro-ids.org