

CIS/TCOM 551

Computer and Network Security

Spring 2005

Lecture 10

Announcements

- Midterm 2 will (hopefully!) be returned on Thursday
- Final project will be on the web site today.
 - Network Intrusion Detection
 - Due last day of classes: Friday, April 22nd.
 - Groups of 2 or 3 students
 - E-mail Eric Cronin ecronin@cis.upenn.edu by Friday.

Today's Agenda

- Revisit Network Intrusion Detection Systems
- Reading: "Bro: A System for Detecting Network Intruders in Real-Time" by Vern Paxson
 - Available on web pages
 - Some of today's slides are taken from a talk by Paxson
- Project 3 is to build a simplified NIDS
 - More later

Intrusion Detection Systems

- Network-based IDSs:
 - Use raw packet data to look for possible attacks
 - Run network adapter in promiscuous mode
- Host-based IDSs:
 - Log OS specific events (e.g. system calls)
- Looking for:
 - Known attack patterns (e.g. versus SMTP or port scans)
 - Frequency statistics (to detect DoS)
 - Anomalous behavior (based on some profile of “usual” behavior)

- Example: ZoneAlarm software can detect port scans.

Intrusion Detection

- Known attack signatures
 - As for worms and viruses
- Anomaly detection:
 - Build a profile of “typical” or “normal” behavior
 - What programs are usually running
 - Where do people usually log in from
 - What is the normal timing behavior of your keystrokes
 - How do you arrange your computer desktop...

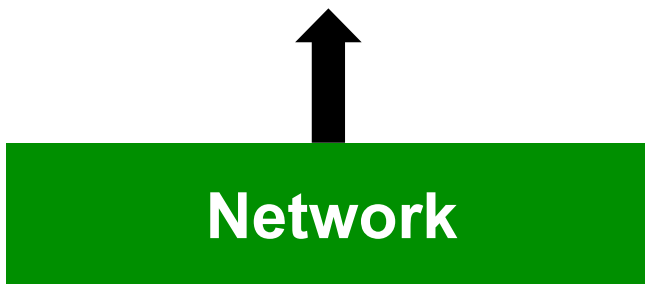
Commercial ID Systems

- ISS – Real Secure from Internet Security Systems:
 - Real time IDS.
 - Contains both host and network based IDS.
- Tripwire – File integrity assessment tool.
- Bro and Snort – open source public-domain system.

Bro: Real time IDS

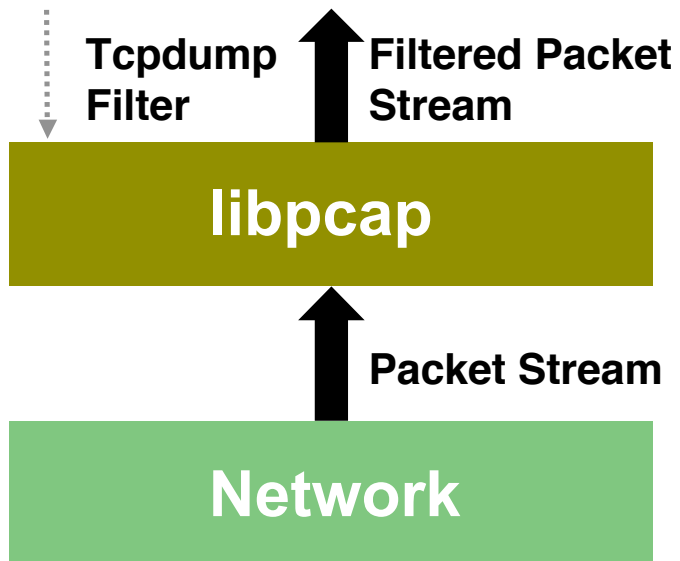
- Network based IDS
- Currently developed for six Internet applications: FTP, Finger, Portmapper, Ident, Telnet and Rlogin.
- Design Goals
 - High-speed, large volume monitoring
 - No packet filter drops
 - Real time notification
 - Mechanism separate from policy
 - Extensible
 - Resilience to attack

How Bro Works



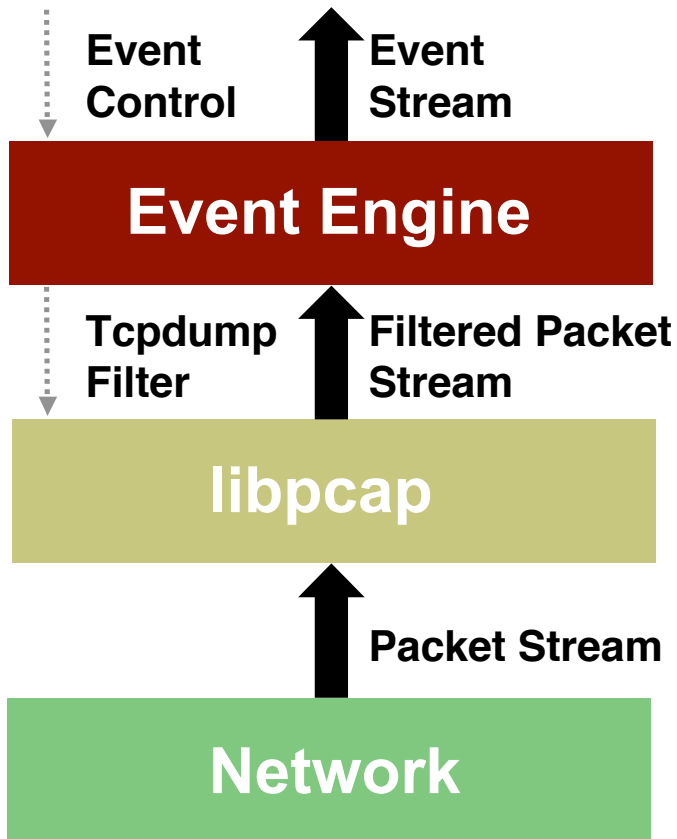
- Taps GigEther fiber link passively, sends up a copy of all network traffic.

How Bro Works



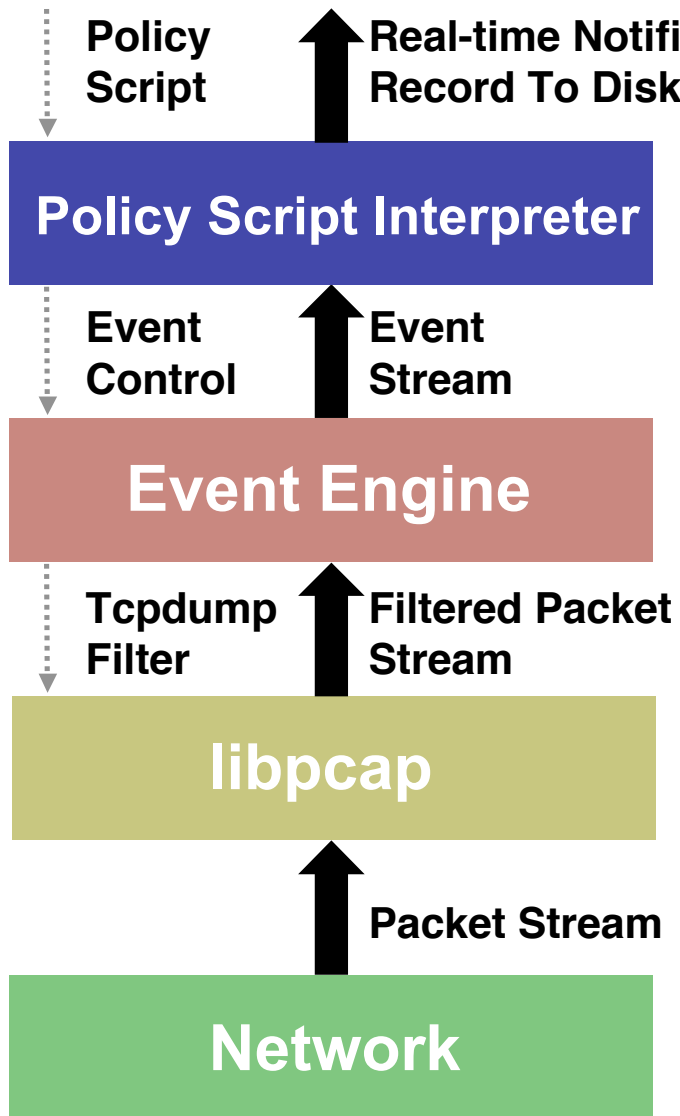
- Kernel filters down high-volume stream via standard *libpcap* packet capture library.

How Bro Works



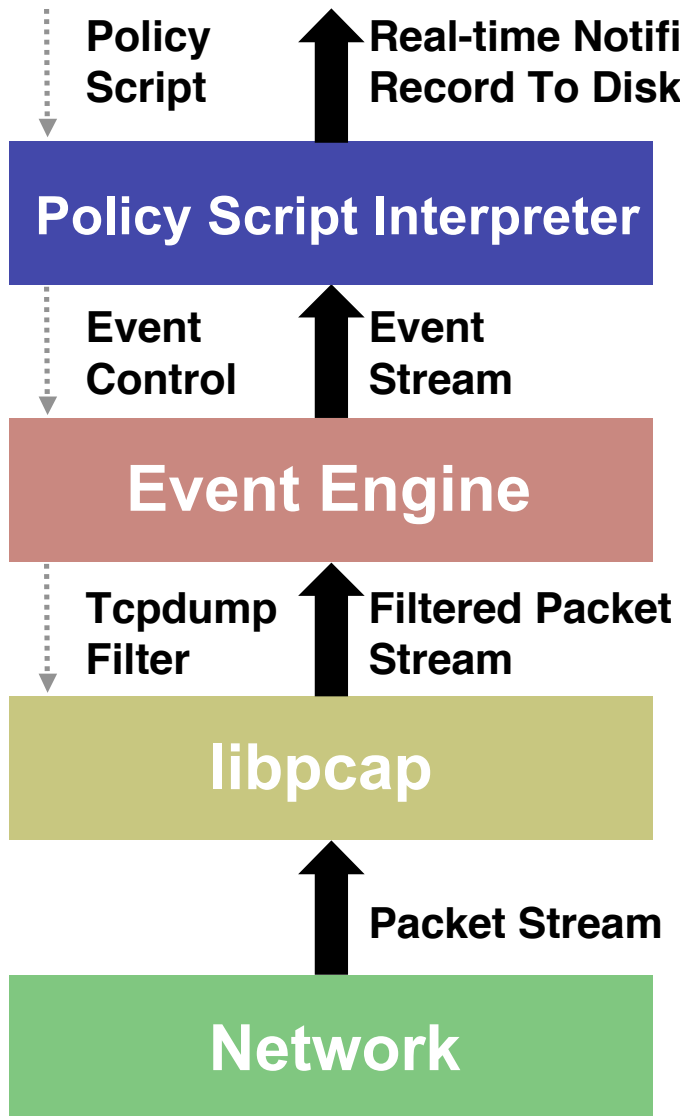
- “Event engine” distills filtered stream into high-level, *policy-neutral* events reflecting underlying network activity
 - E.g. Connection-level:
 - connection attempt
 - connection finished
 - E.g. Application-level:
 - ftp request
 - http_reply
 - E.g. Activity-level:
 - login success

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 - libpcap

- It's the packet capture library used by tcpdump.
- Isolates Bro from details of the network link technology.
- Filters the incoming packet stream from the network to extract the required packets.
 - E.g port finger, port ftp, tcp port 113 (Ident), port telnet, port login, port 111 (Portmapper).
 - Also captures packets with the SYN, FIN, or RST Control bits set.
- You'll get some experience with pcap in Project 3

Bro – Event Engine

- The filtered packet stream from the libpcap is handed over to the Event Engine.
- Performs several integrity checks to assure that the packet headers are well formed.
- It looks up the connection state associated with the tuple of the two IP addresses and the two TCP or UDP port numbers.
 - This is so that Bro can maintain contextual information
- It then dispatches the packet to a handler for the corresponding connection.

Bro – TCP Handler

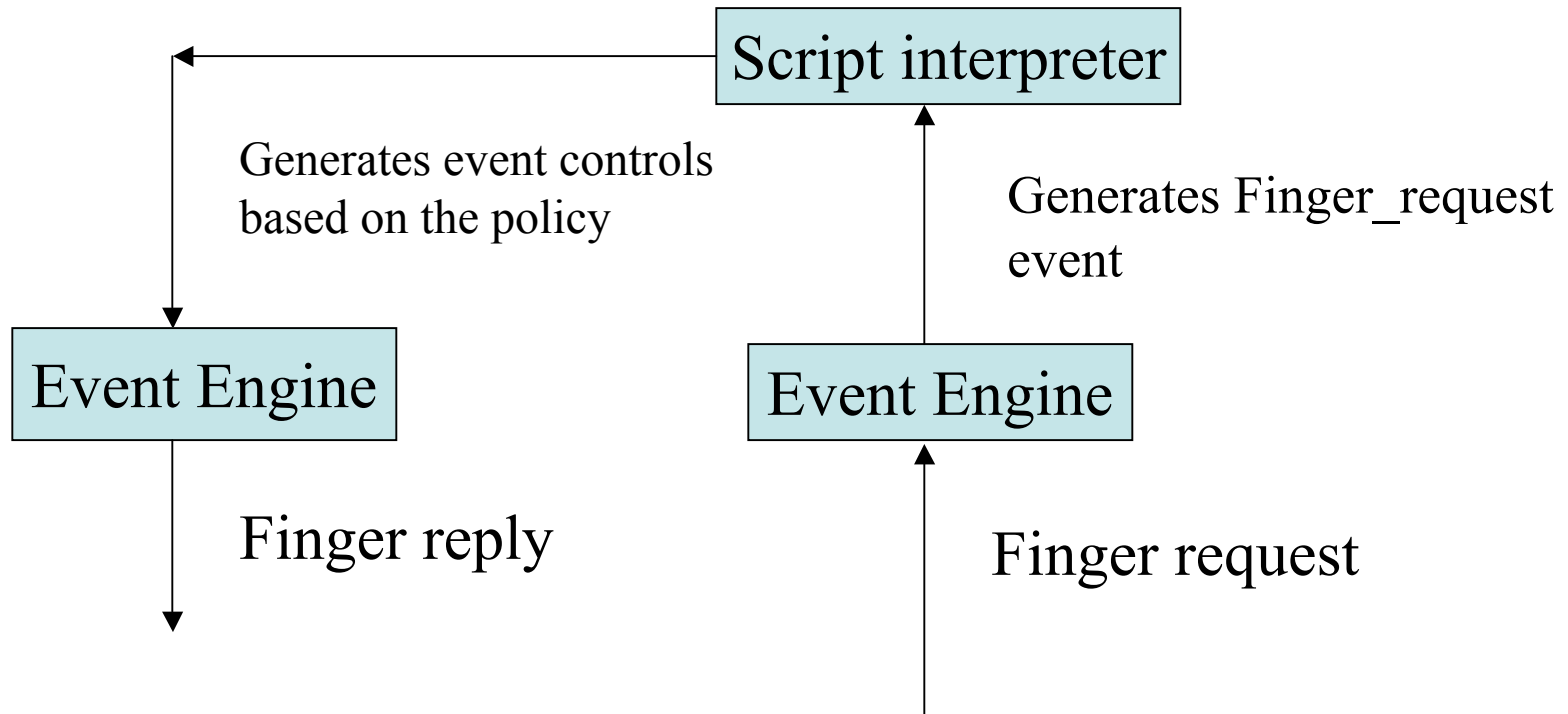
- For each TCP packet, the connection handler verifies that the entire TCP Header is present and validates the TCP checksum.
- If successful, it then tests whether the TCP header includes any of the SYN/FIN/RST control flags and adjusts the connection's state accordingly.
- Different changes in the connection's state generate different events.
 - `connection_attempt`
 - `connection_established`
 - `connection_finished`

Policy Script Interpreter

- The policy script interpreter receives the events generated by the Event Engine.
- It then executes scripts written in the Bro language which generates events like logging real-time notifications, recording data to disk or modifying internal state.
 - Bro language is a domain specific programming language for describing how to handle events
 - See paper for details
- Adding new functionality to Bro consists of adding a new protocol analyzer to the event engine and then writing new events handlers in the interpreter.

Application Specific Processing:Finger

Tests for buffer overflow,
checks the user against
sensitive ids, etc



Application Specific Processing: Telnet

- Try to determine usernames from telnet session information
- Reconstruct events in the Telnet protocol
- Recognize the authentication dialog
- Reconstruct keystrokes
 - Can be tricky due to 'backspace' and 'delete' characters
 - For example: "rboot" might correspond to "root"

Attacks against Bro: Overload

- Overload attack
 - Attacker swamps the monitor with packets (DoS)
 - Typical strategy is to shed load (e.g. selectively drop some streams)
 - If attacker knows strategy for shedding load, may take advantage of that information
- Defenses:
 - Use sufficient hardware to monitor max rate of link traffic
 - Bro generates `net_stat_update` events every few seconds to allow high-level policy scripts to react to changes in load

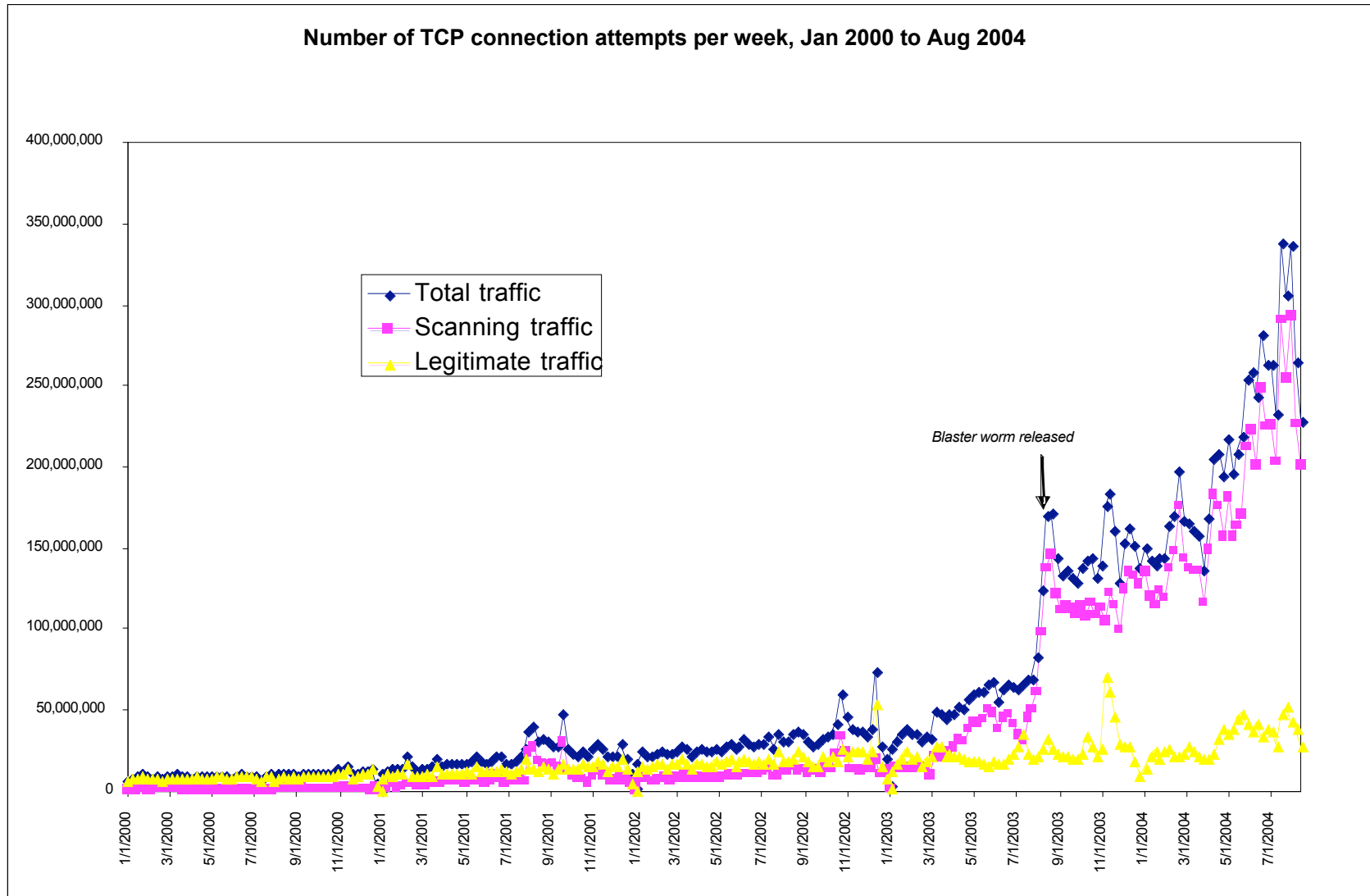
Attacks against Bro: Crashes

- Cause monitor to crash due to bugs/coding error
 - Assume that source code of Bro is available to the attacker but policy script is not.
- Exhaust monitor resources
 - Create traffic that consumes a lot of memory
- Defenses:
 - Unix alarm for a "watch dog" timer that checks to see whether the event engine has failed to handle packet after T seconds. Allows to detect and recover from failure of monitor.
 - The shell script that runs Bro checks for unexpected termination, and if so, uses `tcpdump` to log the traffic (to allow for detection & recovery)

Attacks against Bro: Subterfuge

- Mislead the monitor about the traffic it sees.
 - Hard to detect
 - Example: Embed \0 into strings at unexpected places
- Whole can of worms that we'll talk about more

LBNL Inbound TCP Traffic



LBNL Inbound TCP Traffic

