September 15, 1998
RAID 98

vern@ee.lbl.gov

University of California, Berkeley
Lawrence Berkeley National Laboratory

Network Research Group

Vern Paxson

Experiences and Status
Using Bro to Detect Network Intruders:
Design goals & constraints:

- High-speed, large volume monitoring (FDDI).
- No packet filter drops.
- Real-time notification.
- Mechanism separate from policy.
- Extensible.
- Avoid simple mistakes ↔ special-purpose policy language.

The monitor will be attacked.
Network

Packet stream

Libpcap

Tcpdump filter
Filtered packet stream

Event Engine

Event control
Event stream

Policy Script Interpreter

Policy script

Record to disk

Real-time notification
The Bro Language:

- **Strongly typed** catch errors at compile time.
- Usual arithmetic (including count, bool).
- Strings are counted rather than NUL-terminated.
- User: ncte/owser.root
- Addr: IP address (e.g., "128.3.254.23", TP1.gov[*]).
- Port: TCP or UDP port (e.g., "http", 80/tcp).
- Time, Interval types (and constants, e.g., "30 min").

Strings are counted rather than NUL-terminated:

- The Bro Language:
The **Bro** language, contains record, packets, parse-ftp-port, etc.

**Record** types for a collection of values.

**Table** types for associative arrays.

**Table**, types for a collection of values.

Built-in functions: fmt, edit, mask, addr, network, time, etc.

const allowed-services = set

```
{{ [ntp.ttl.gov, nntp]
    [ftp-sever, ftp, smtp, auth, smtp, auth, 20/tcp]
} = [addr, port] =
```

Easy to construct large tables:

(e.g. `src-addr`, `dst-addr`, `serv` in RPC-

```
The usual control constructs (except no loops), plus `log`.

Event handlers look just like function definitions w/ `function` replaced by `event`.

Events are queued FIFO, drained after each new packet.

Event generators an event — looks just like a function call.

Bro statements:
Attacks on the monitor:

- Initiation, retransmit attack data with insufficient TTL.
- E.G., send innocuous data with insufficient TTL to reach des-
- s.
- Subterfuge: fool the monitor.
  - E.g., send innocuous data with insufficient TTL to reach des-
  - s.

Crash:
- Make it fault, or run out of resources.
- Defense: watchdog timer & backup tracing; resource man-

Overload:
- Make it drop packets, sneak in.
- Defense: leave doubt as to monitor's capabilities; shed load.
- Defense: leave doubt as to monitor's capabilities; shed load.
- Data loss, & leave doubt as to monitor's capabilities.
  - Subterfuge: fool the monitor.

Defense:
  - Lea de doubt as to monitor's capabilities; shed load.
  - Overload: make it drop packets, sneak in.
  - Subterfuge: fool the monitor.

Attacks on the monitor:

- Initiation, retransmit attack data with insufficient TTL.
- E.G., send innocuous data with insufficient TTL to reach des-
- s.
- Subterfuge: fool the monitor.
  - E.g., send innocuous data with insufficient TTL to reach des-
  - s.

Crash:
- Make it fault, or run out of resources.
- Defense: watchdog timer & backup tracing; resource man-

Overload:
- Make it drop packets, sneak in.
- Defense: leave doubt as to monitor's capabilities; shed load.
- Defense: leave doubt as to monitor's capabilities; shed load.
- Data loss, & leave doubt as to monitor's capabilities.
  - Subterfuge: fool the monitor.

Defense:
  - Lea de doubt as to monitor's capabilities; shed load.
  - Overload: make it drop packets, sneak in.
  - Subterfuge: fool the monitor.
Subterfuge attacks, cont.

In part, exercise great care when developing IDS, attending to possible exceptional events. Development of Bro attempted.

But perhaps hopeless, due to basic ambiguities.

See the excellent paper "Vulnerabilities in Network Intrusion Detection Software," Ptacek/Newsham:

http://www.securenetworks.com/papers/IDS.PS
Application analysis:

For all TCP connections:
- start time, duration, service, addresses, sizes
- port, address scanning

 attempts and requests/replies.

 XDR on top of UDP & TCP.

Portmapper: set/unset/getport/dump/callit

Finger: finger-request (request with request string)
HTTP: http-request, http-status (telenging).

Heuristic: activatation-encryption, telnet-confused

authentatcation-rejected

authentatcation-rejected

authentatcation-accepted

Telnet: Telnet-logged-in, Telnet-failure

Script parses PORT, PASY to identify subsequent conn.

FTP: ftp-request, ftp-reply.
Can sustain > 1,250 filtered packets/sec (peak 5,000/sec).

But: filter discards a whole lot.

Currently: 400 MHz Pentium II, plenty of disk & memory.

25 Mbps-busy-hour-utilization FDDI ring no problem.

Freely available with sample policy scripts.

Digital Unix, FreeBSD, IRIX, Linux, SunOS, Solaris.

Status:

25,000 lines C++, 1,900 lines of policy script.

（but no documentation :-( )
Status, cont:
In operation since April, 1996.

- 800,000,000 packets per day.
- 60 MB connections logs per day.
- 30 real-time notifications per day (large variance).
- Scans detected: everyday.
- Break-ins detected: not uncommon.
- 6,000 email messages.
- 100+ CERT/CIAC incident reports.
TCPs that ack data that was never sent.

TCPs that retransmit different (valid) data than the first time.

Crud seen on a DMZ:

Storms of 10,000+ FIN or RST packets.

SYN packets with URG set.

Private addresses leaking out.

Fragments with DF set.

Legitimate tiny fragments.

Overlapping fragments.

TCP connections reused prematurely.
For now, record all LBNL interactive traffic.
Can copy full FDDI stream to disk w/o Drops.
Record everything: striped disks → single 69 GB partition.

Soon: LAN monitoring.

- Record everything.
- Telnet analysis for full FDDI ring (20,000+/day)
- Spot tracking
- Bro HTTP (planned)
- Bro except for HTTP

Five monitor machines on DMZ:

Additioanl monitoring:
Near-term future directions:

More application modules.

Regular expression matching.

Automatic demuxing of “hot” streams.

Generic spot-tracing tool.

Reactive firewall.
Longer-term future directions:

Wish list has 100+ items.

BPF speed hacks.

Compiling and optimizing scripts.

Distributing across multiple CPUs/hosts.

More application modules.

Wish list has 100+ items.
Includes link to BRO USENIX paper.


Further info:

Send mail to bro-request@lbl.gov, subject "subscribe"

Bro mailing list: bro@lbl.gov

BSD copyright (but please don't redistribute alpha releases)

ftp://ftp.ee.lbl.gov/vp-bro-0.4-alpha.tar.gz

Current BRO 0.4 alpha release:
Acknowledgments:

Many thanks to DEC-WRL (Jeff Mogul in particular) for ERP grant of monitoring hardware.

Loken and Dave Stevens.
And to Craig Leres, Mark Rosenberg, Van Jacobson, Stu