Intrusion Detection

EECE 571B Computer Security

Konstantin Beznosov

Intrusion Characteristics

- Main idea: a *compromised* system has different characteristics than a normal one
  - Statistical patterns of activity
  - Suspicious activity
  - Specifications

IDS goals

- Detect wide range of intrusions
  - Including previously unknown attacks
- Detect intrusions quickly
  - Allow timely response
  - A good IDS can be used for intrusion *prevention*
- Explain intrusions well
  - Allow intelligent response
- Detect accurately

intrusion detection strategies

- signature detection
  - decide in advance what type of behavior is undesirable (security policy)
  - codify undesirable behavior into signatures
  - promises to detect intrusions in a timely and efficient manner
  - problems
    - attacks and violations have to be easily codified into signatures (security policies)
    - **difficulty in detecting previously unknown intrusions**
    - intrusion signatures must be updated frequently
- anomaly detection
  - declare everything that is unusual for the subject suspect, and rise an alarm
  - promises to detect
    - abuses of legitimate privileges that cannot easily be codified into **security policy**
    - detect attacks that are “novel” to the intrusion detection system
  - problems
    - tendency to **take up data processing resources**
    - the possibility of an **attacker teaching the system** that his illegitimate activities are ordinary

desirable properties of IDSs

- effectiveness
  - to what degree does it detect intrusions into the target system, and how good is it at rejecting false positives (false alarms)?
- efficiency
  - the run-time efficiency of the intrusion detection system, how many computing resources and how much storage does it consume, can it make its detections in real-time?
- ease of use
  - **How easy is it to field and operate for a user who is not a security expert? What demands can be made of the person responding to the intrusion alarm? How high a false alarm rate can he/she realistically be expected to cope with, and under what circumstances is he/she likely to ignore an alarm?**
- security
  - ability to sustain attacks on IDS itself
- interoperability with other IDSs
- transparency
  - how disruptive for an organization deployment and operation of an IDS
- collaboration with other security (mechanisms) in the system/network

health example

- the **basic rate of incidence** is only 1/10,000 = P(S)
- test is 99% accurate
  - P(R|S) = 99% and P(¬R|¬S) = 99%
- you tested positive for the disease *(R)*
- what's the probability *P(S|R)* of you having the disease?

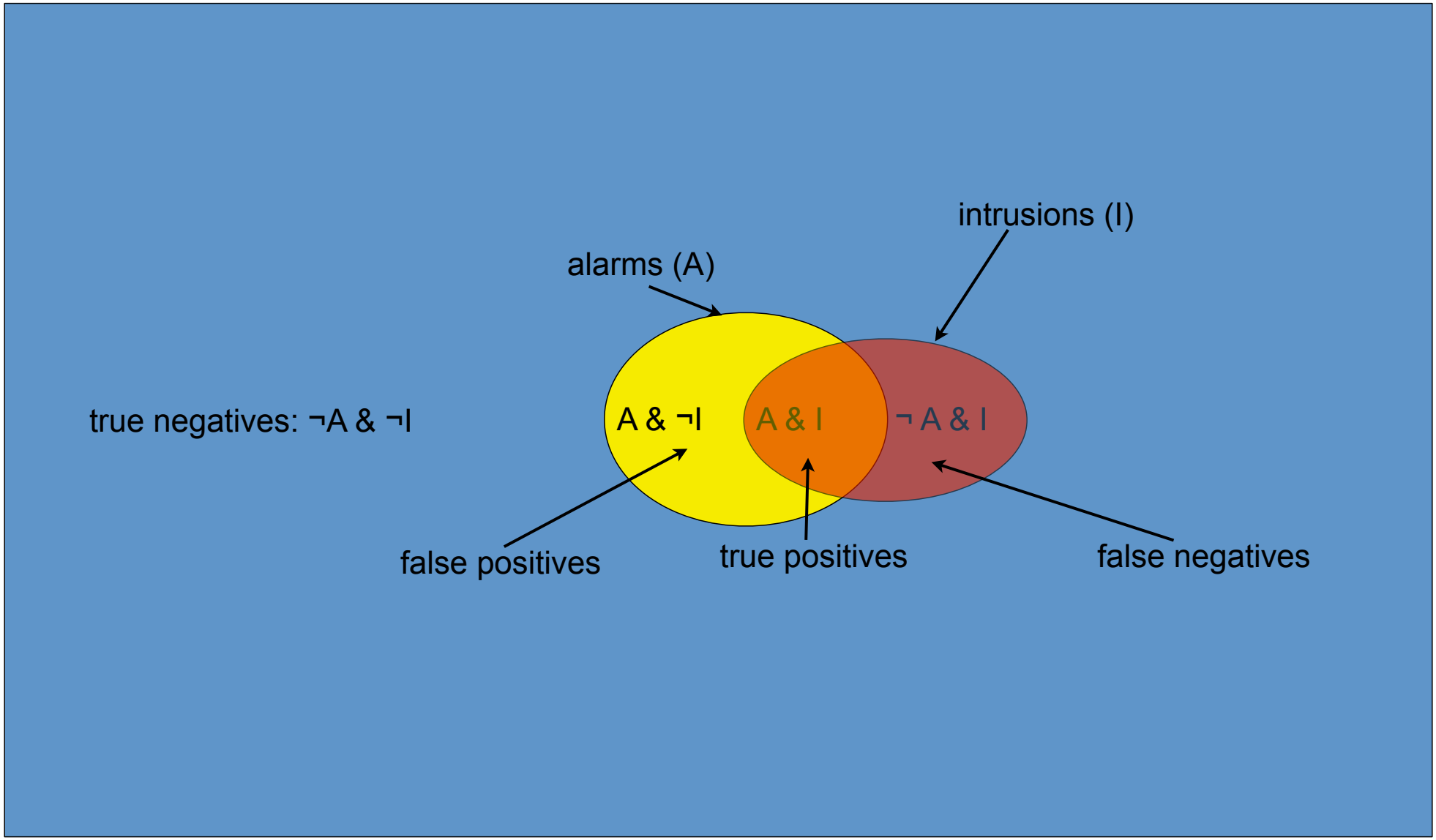
$$P(S|R) = \frac{P(S) \cdot P(R|S)}{P(S) \cdot P(R|S) + P(\neg S) \cdot P(R|\neg S)}$$

$$P(S|R) = \frac{1/10000 \cdot 0.99}{1/10000 \cdot 0.99 + (1 - 1/10000) \cdot 0.01} = 0.00980\ldots \approx 1\%.$$

adopted from [3]

the base-rate fallacy

Venn diagram for ID event space



let's plug some IDS numbers

- 1,000,000 audit records per day
- 1-2 intrusions per day
- 10 records per event (including intrusion)
- one site security officer
  - can only react to low number of alarms
  - false alarm rate should be less than 50%

Bayesian detection rate

$$P(I|A) = \frac{P(I) \cdot P(A|I)}{P(I) \cdot P(A|I) + P(\neg I) \cdot P(A|\neg I)}$$

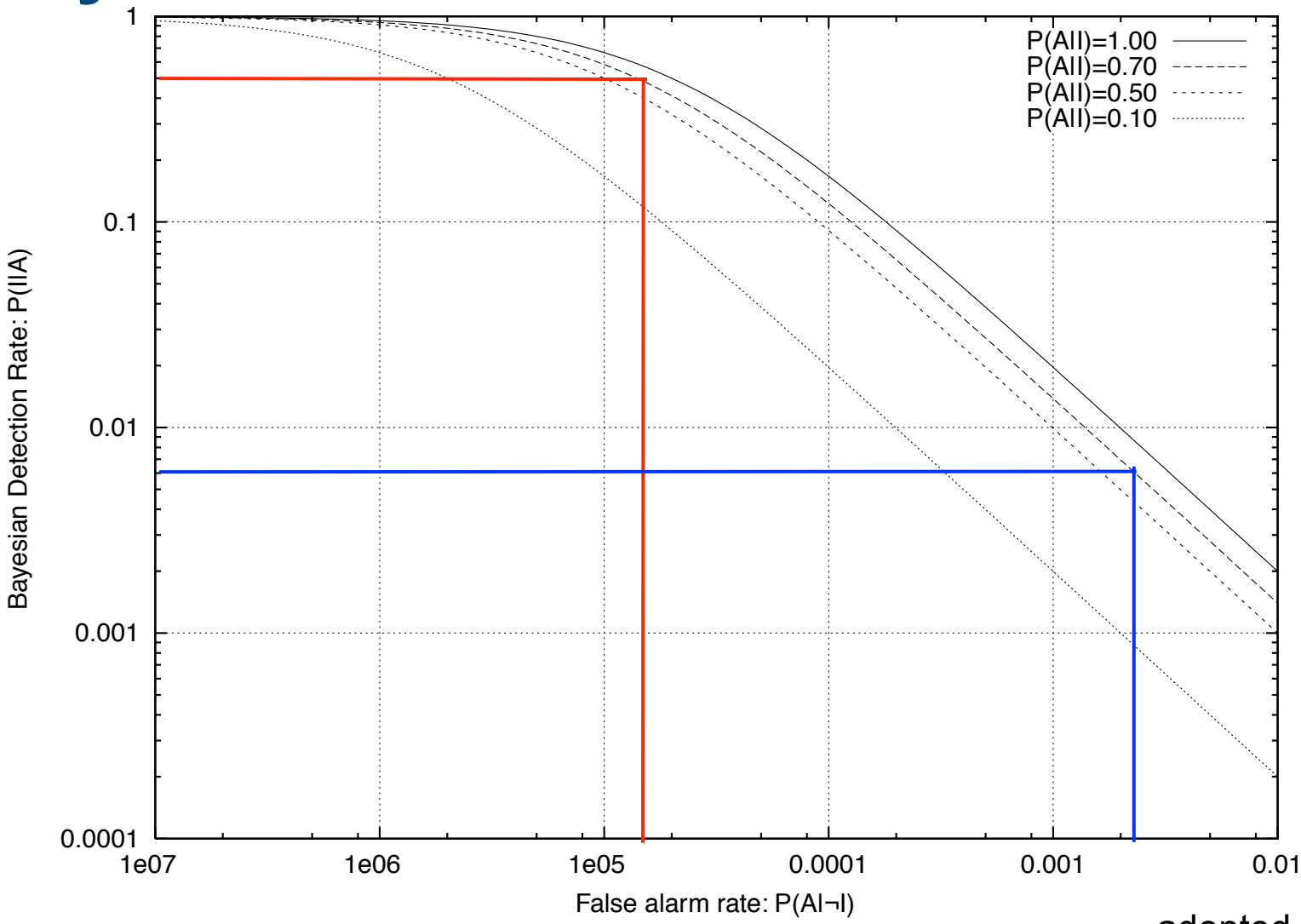
$$P(I) = 1 \Big/ \frac{1 \cdot 10^6}{2 \cdot 10} = 2 \cdot 10^{-5}; P(\neg I) = 1 - P(I) = 0.99998$$

$$P(I|A) = \frac{2 \cdot 10^{-5} \cdot P(A|I)}{2 \cdot 10^{-5} \cdot P(A|I) + 0.99998 \cdot P(A|\neg I)}$$

adopted from [3]

So what?
Even for the unrealistically high detection rate 100%, we have to have a very low false alarm rate (on the order of 10^{-5}) for the Bayesian detection rate to have a value of 66%

Bayesian detection vs. false alarm rate



adopted from [3]

conclusions from IDS base-rate fallacy

- "**the factor limiting the performance of an intrusion detection system** is not the ability to identify behavior correctly as intrusive, but rather **its ability to suppress false alarms**"
- "one should measure **the false alarm rate in relation to how many intrusions one would expect to detect**, not in relation to the maximum number of possible false alarms"

Anomaly Models

- Manual models
  - Describe what behavior is correct or anomalous
- Statistical models
  - Learn what is the normal behavior

Statistical Models

- Monitor system in normal state
- Learn patterns of activity
  - Various statistical models to do this
- Decide an intrusion threshold
  - E.g. 2 standard deviations from normal
- Adapt over time (optional)

Simple Model (Normal)

- Measure values of parameters
  - e.g., network load
- Calculate mean & standard deviation
- Set a threshold based on a confidence interval
  - e.g., 2 standard deviatons =~ 95%
  - 3 standard deviations =~ 99.7%
- Alert for values outside the threshold

Markov Models

- Consider anomalous *sequences* of operations
  - Usually system calls
- Markov models: next operation depends on current one
  - E.g. read follows open
- Transition probabilities computed by training
- Can classify likelihood of sequences

Higher Order Markov Models

- First order Markov models consider only the previous state
  - I.e. likelihood of each digram of operations
  - E.g. if training set is:
    - how is it going?
    - the sky is blue.
  - Then the sentence “how is blue” falls within the model
- Higher order Markov models consider several previous states

n-grams

- Another way to think about previous states is with n-grams

  open read write open mmap write fchmod close

- 3-grams are:

  open read write
  write open mmap
  mmap write fchmod
  fchmod close
  read write open
  open mmap write
  write fchmod close

Statistical Models

- Pro:
  - No need to know what is “normal” in advance
  - Flexibility between installations
  - Adaptive
  - Control of false positive rates

Statistical Models

- Cons:
  - Statistical model may be wrong
    - E.g. not normally distributed data
  - Training set may be inadequate
    - Same problem as testing
  - Alerts difficult to explain
  - Attacks may be able to get around them

Misuse specification

- Look for patterns of activity that *shouldn't* happen
  - e.g., control transfer to a randomized location
  - e.g., traffic with internal address coming from outside
- Usually very low false positive rate
- But only detects known attacks

Specification-based Detection

- Specify correct operation, everything else an attack
- E.g. *rdist* specification
  - open world readable files
  - open non-world readable files rdist creates
  - create files in /tmp
  - chown/chmod files it creates
- Any other filesystem operation is an error

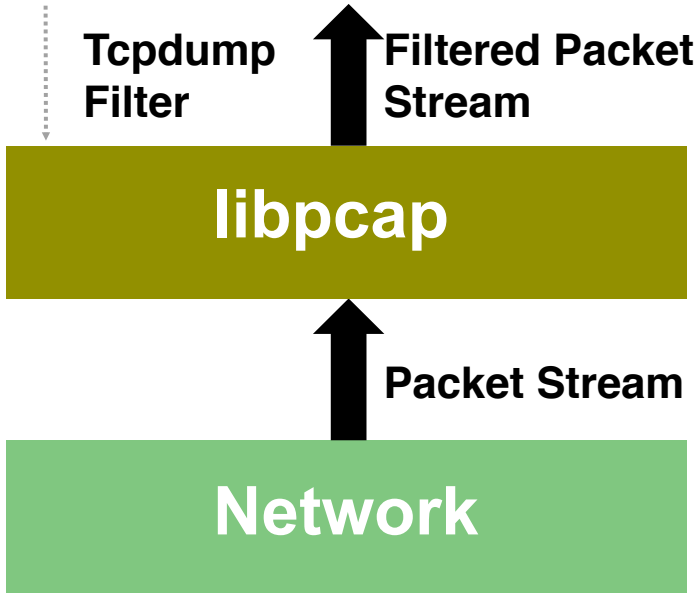
How Bro Works

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

Bro

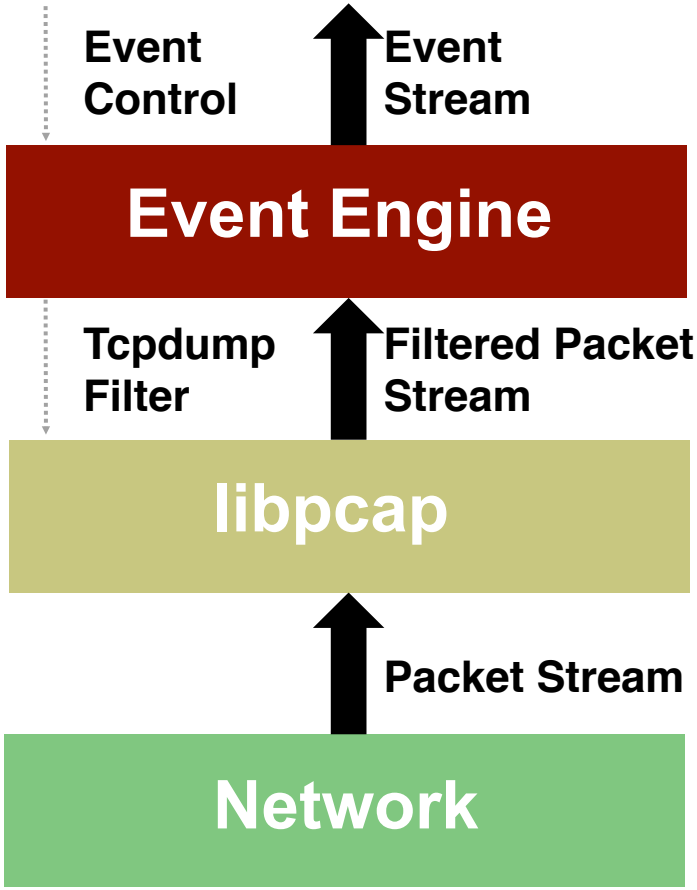
adopted from [2]

How Bro Works



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

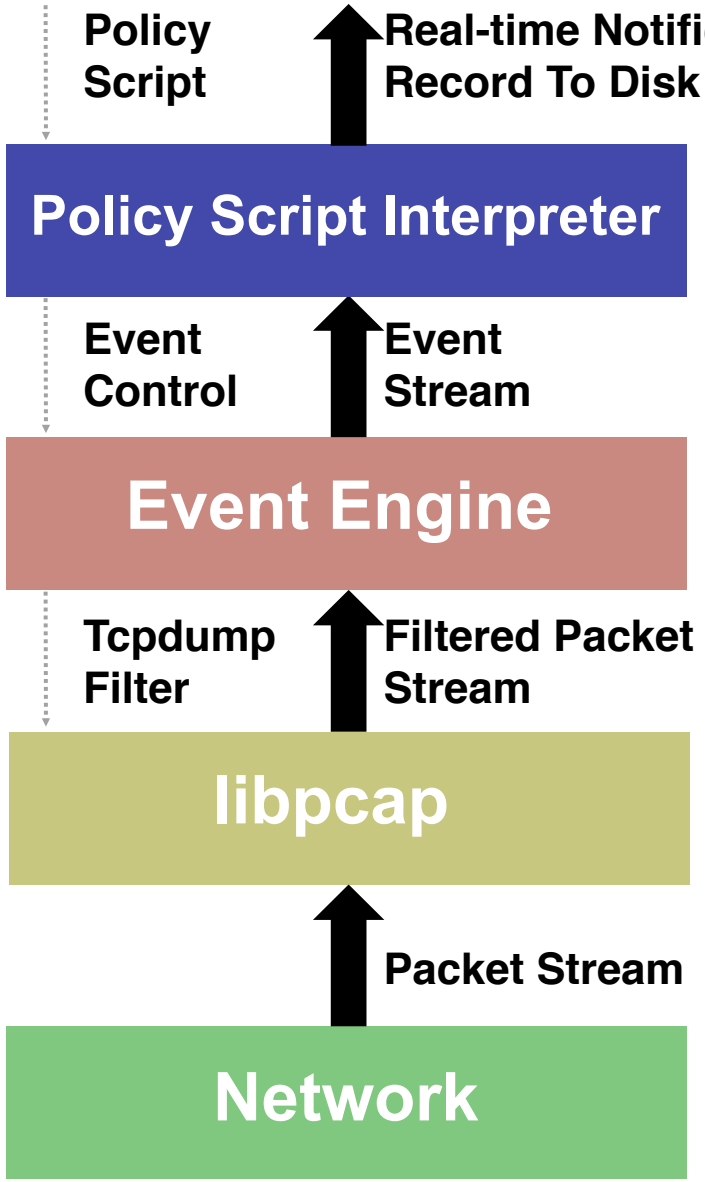
How Bro Works



- "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

adopted from [2]

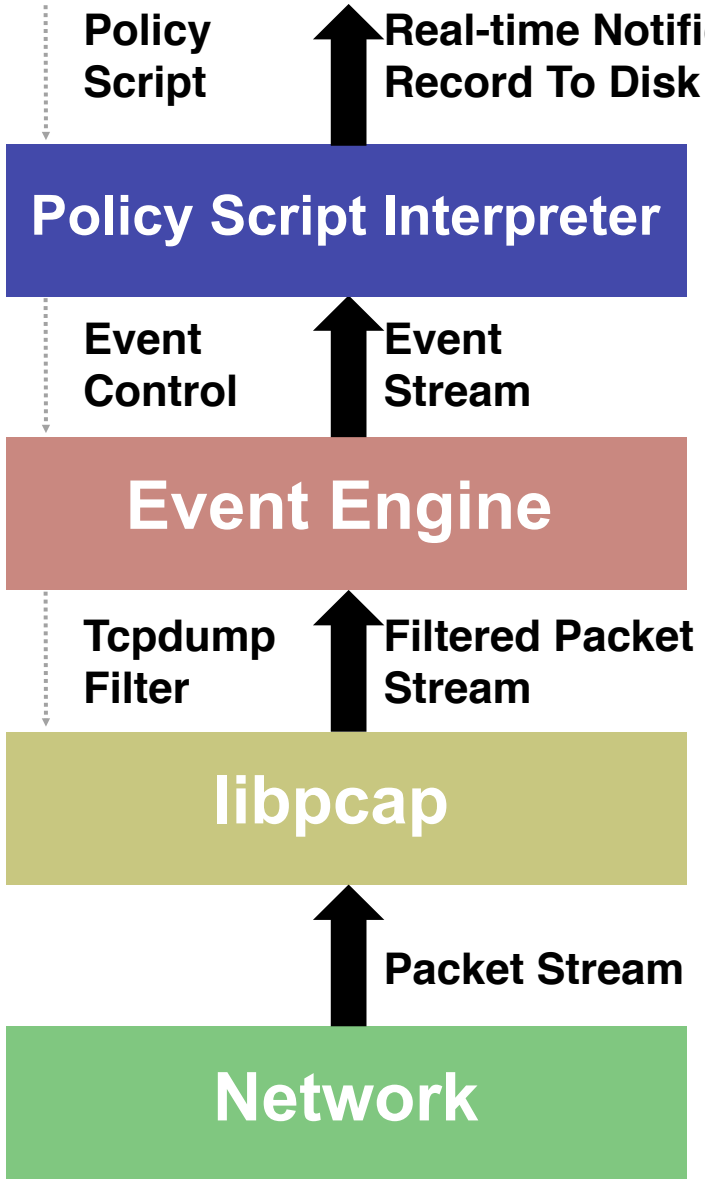
How Bro Works



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

adopted from [2]

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*, email, paging, etc.
  - Executes programs as a form of response

adopted from [2]

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);
```

Mimicry Attacks

- Tailor attack specifically to an IDS
- e.g., pad system calls sequences to look legitimate
- Normal sequence:

  open read write close open fchmod close exec
- Naïve attack:

  open read exec
- Mimicry attack (digrams):

  open read write close exec

Network Intrusion Detection

- Most attacks come from the outside network
- Monitoring outside link(s) easier than monitoring all systems in an enterprise
- Network Intrusion Detection Systems (NIDS) a popular tool

NIDS challenges

- NIDS Challenges
  - Volume of traffic
  - Attacks on the monitor
  - Uncertainty about host behavior

Intrusion Response

- Once intrusion is detected, what to do?
- Prevention
  - Stop the attack if detected fast enough
- Containment
  - Prevent further damage
- Eradication
  - Restore system to known good state
- Follow-Up
  - Track down attackers
- Most work is on eradication

credits

These slides incorporate some of the material from

1. “Intrusion Detection” course CS463.12 at the University of Illinois at Urbana-Champaign
2. B. L. Tierney, V. Paxson, “An Overview of the Bro Intrusion Detection System,” presentation.
3. S. Axelsson, “The base-rate fallacy and the difficulty of intrusion detection,” ACM Trans. Inf. Syst. Secur. 3, 3 (August 2000), 186-205