CSE 484 / CSE M 584 (Autumn 2011)

Security and Networks

Daniel Halperin
Tadayoshi Kohno

Thanks to Dan Boneh, Dieter Gollmann, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...
Class updates

• Homework 3 due today
• My office hours this week:
  • **CSE 210:** W, Th, F in the after-class slot
  • Other times by appointment.
• Come pick up graded Homework #2
Lab 3

- Posted on website and on Catalyst.
  - https://catalyst.uw.edu/collectit/assignment/dhalperi/17513/72548
  - Hack my privacy!

- This lab is optional
  - Can only help your grade.
  - Lots of opportunity for extra credit.
  - I really think this lab is fun, and encourage you to do it, but we’re not going to require it.
This week

- **Today:** Finish networks, Final, & Course Evals
- **Friday:** Any questions you have
  - Submit to my email, cse484-tas
  - Submit anonymously via the feedback form on the website
Grading?
Final?
SYN Flooding Attack

Listening...
Spawn a new thread, store connection data
... and more
... and more
... and more
... and more
SYN Flooding Explained

- Attacker sends many connection requests with spoofed source addresses
- Victim allocates resources for each request
  - Connection state maintained until timeout
  - Fixed bound on half-open connections
- Once resources exhausted, requests from legitimate clients are denied
- This is a classic denial of service (DoS) attack
  - Common pattern: it costs nothing to TCP initiator to send a connection request, but TCP responder must allocate state for each request (asymmetry!)
Preventing Denial of Service

- **DoS** is caused by asymmetric state allocation
  - If responder opens a state for each connection attempt, attacker can initiate thousands of connections from bogus or forged IP addresses
- **Cookies** ensure that the responder is stateless until initiator produced at least 2 messages
  - Responder’s state (IP addresses and ports of the connection) is stored in a cookie and sent to initiator
  - After initiator responds, cookie is regenerated and compared with the cookie returned by the initiator
SYN Cookies

Does not store state

Client should not be able to invert a cookie (why?)

Cookie must be unforgeable and tamper-proof (why?)

Recompute cookie, compare with with the one received, only establish connection if they match

More info: http://cr.yp.to/syncookies.html
Anti-Spoofing Cookies: Basic Pattern

- Client sends request (message #1) to server
- Typical protocol:
  - Server sets up connection, responds with message #2
  - Client may complete session or not (potential DoS)
- Cookie version:
  - Server responds with hashed connection data instead of message #2
  - Client confirms by returning hashed data
    - If source IP address is bogus, attacker can’t confirm
  - Need an extra step to send postponed message #2, except in TCP (SYN-ACK already there)
Another Defense: Random Deletion

 SYN\textsubscript{C} \rightarrow \text{half-open connections}

\begin{tabular}{|c|}
\hline
121.17.182.45 \hline
231.202.1.16 \hline
121.100.20.14 \hline
5.17.95.155 \hline
\end{tabular}

- If SYN queue is full, delete random entry
  - Legitimate connections have a chance to complete
  - Fake addresses will be eventually deleted
- Easy to implement
“Ping of Death”

- If an old Windows machine received an ICMP packet with a payload longer than 64K, machine would crash or reboot
  - Programming error in older versions of Windows
  - Packets of this length are illegal, so programmers of Windows code did not account for them

- Recall “security theme” of this course - every line of code might be the target of an adversary

Solution: patch OS, filter out ICMP packets
Intrusion Detection Systems

- Advantage: can recognize new attacks and new versions of old attacks

- Disadvantages
  - High false positive rate
  - Must be trained on known good data
    - Training is hard because network traffic is very diverse
  - Definition of “normal” constantly evolves
    - What’s the difference between a flash crowd and a denial of service attack?
Intrusion Detection Problems

- Lack of training data with real attacks
  - But lots of “normal” network traffic, system call data

- Data drift
  - Statistical methods detect changes in behavior
  - Attacker can attack gradually and incrementally

- Main characteristics not well understood
  - By many measures, attack may be within bounds of “normal” range of activities

- False identifications are very costly
  - Sysadm will spend many hours examining evidence
Intrusion Detection Errors

 salvar

- **False negatives**: attack is not detected
  - Big problem in signature-based misuse detection

- **False positives**: harmless behavior is classified as an attack
  - Big problem in statistical anomaly detection

- Both types of IDS suffer from both error types

- Which is a bigger problem?
  - Attacks are fairly rare events
Base-Rate Fallacy

1% of traffic is SYN floods; IDS accuracy is 90%
- IDS classifies a SYN flood as attack with prob. 90%,
  classifies a valid connection as attack with prob. 10%

What is the probability that a connection flagged by IDS as a SYN flood is actually valid traffic?
Suppose two events A and B occur with probability $\Pr(A)$ and $\Pr(B)$, respectively.

Let $\Pr(AB)$ be probability that both A and B occur.

What is the conditional probability that A occurs assuming B has occurred?
Suppose two events A and B occur with probability \( \Pr(A) \) and \( \Pr(B) \), respectively.

Let \( \Pr(AB) \) be probability that both A and B occur.

What is the conditional probability that A occurs assuming B has occurred?

\[
\Pr(A \mid B) = \frac{\Pr(AB)}{\Pr(B)}
\]
Suppose mutually exclusive events $E_1, \ldots, E_n$ together cover the entire set of possibilities. Then probability of any event $A$ occurring is

$$Pr(A) = \sum_{1 \leq i \leq n} Pr(A \mid E_i) \cdot Pr(E_i)$$

- Intuition: since $E_1, \ldots, E_n$ cover entire probability space, whenever $A$ occurs, some event $E_i$ must have occurred.

Can rewrite this formula as

$$Pr(E_i \mid A) = \frac{Pr(A \mid E_i) \cdot Pr(E_i)}{Pr(A)}$$
1% of traffic is SYN floods; IDS accuracy is 90%
- IDS classifies a SYN flood as attack with prob. 90%,
  classifies a valid connection as attack with prob. 10%

What is the probability that a connection flagged by IDS as a SYN flood is actually valid traffic?
1% of traffic is SYN floods; IDS accuracy is 90%.

- IDS classifies a SYN flood as attack with prob. 90%,
  classifies a valid connection as attack with prob. 10%

What is the probability that a connection flagged by IDS as a SYN flood is actually valid traffic?

\[
Pr(\text{valid} | \text{alarm}) = \frac{Pr(\text{alarm} | \text{valid}) \cdot Pr(\text{valid})}{Pr(\text{alarm})}
\]
1% of traffic is SYN floods; IDS accuracy is 90%:
- IDS classifies a SYN flood as attack with prob. 90%,
  classifies a valid connection as attack with prob. 10%

What is the probability that a connection flagged by IDS as a SYN flood is actually valid traffic?

\[
Pr(\text{valid} | \text{alarm}) = \frac{Pr(\text{alarm} | \text{valid}) \cdot Pr(\text{valid})}{Pr(\text{alarm})} \\
= \frac{Pr(\text{alarm} | \text{valid}) \cdot Pr(\text{valid}) + Pr(\text{alarm} | \text{SYN flood}) \cdot Pr(\text{SYN flood})}{Pr(\text{alarm})}
\]
Base-Rate Fallacy

1% of traffic is SYN floods; IDS accuracy is 90%
- IDS classifies a SYN flood as attack with prob. 90%,
  classifies a valid connection as attack with prob. 10%

What is the probability that a connection flagged by
IDS as a SYN flood is actually valid traffic?

\[
\Pr(\text{valid | alarm}) = \frac{\Pr(\text{alarm | valid}) \cdot \Pr(\text{valid})}{\Pr(\text{alarm})}
\]

\[
= \frac{\Pr(\text{alarm | valid}) \cdot \Pr(\text{valid}) + \Pr(\text{alarm | SYN flood}) \cdot \Pr(\text{SYN flood})}{0.10 \cdot 0.99 + 0.90 \cdot 0.01}
\]

= \frac{0.10 \cdot 0.99 + 0.90 \cdot 0.01}{0.10 \cdot 0.99 + 0.90 \cdot 0.01}

= \frac{0.099}{0.1091}

= 0.905
Base-Rate Fallacy

- 1% of traffic is SYN floods; IDS accuracy is 90%
  - IDS classifies a SYN flood as attack with prob. 90%, classifies a valid connection as attack with prob. 10%

What is the probability that a connection flagged by IDS as a SYN flood is actually valid traffic?

\[
\frac{\text{Pr(alarm | valid)} \cdot \text{Pr(valid)}}{\text{Pr(alarm)}} = \frac{\text{Pr(alarm | valid)} \cdot \text{Pr(valid)} + \text{Pr(alarm | SYN flood)} \cdot \text{Pr(SYN flood)}}{\text{Pr(alarm | valid)} \cdot \text{Pr(valid)}}
\]

\[
\frac{0.10 \cdot 0.99}{0.10 \cdot 0.99 + 0.90 \cdot 0.01} = 0.92 = 92\% \text{ chance raised alarm is false}!!
\]