disc. 13
cs161 su23

intrusion detection, malware, tor

last discussion :(

slides
bit.ly/cs161-disc

feedback
bit.ly/abhifeedback

hack of the day

hack of the day

- Alibaba Cloud compromised

hack of the day

- Alibaba Cloud compromised
  - “…privilege escalation flaw…and a remote code execution bug…made it possible to elevate privileges to root within the container…and ultimately obtain unauthorized access to the API server.”

hack of the day

- Alibaba Cloud compromised
  - “…privilege escalation flaw…and a remote code execution bug…made it possible to elevate privileges to root within the container…and ultimately obtain unauthorized access to the API server.”
  - all because of a bug in AnalyticDB and AsparaDB!

hack of the day

- Alibaba Cloud compromised
    - “...privilege escalation flaw...and a remote code execution bug...made it possible to elevate privileges to root within the container...and ultimately obtain unauthorized access to the API server.”
    - all because of a bug in AnalyticDB and AsparaDB!
    - 58% of orgs don’t enforce MFA for root users

general questions, concerns, etc.

reminder

please fill out course evaluations!

course-evaluations.berkeley.edu

denial of service (DoS)

denial of service (DoS)

- application level:

denial of service (DoS)

- application level:
    - use inputs that overwhelm the victim in computing the output

denial of service (DoS)

- application level:
    - use inputs that overwhelm the victim in computing the output
    - algorithmic complexity attack

denial of service (DoS)

- application level:
  - use inputs that overwhelm the victim in computing the output
  - algorithmic complexity attack
- network level:

denial of service (DoS)

- application level:
    - use inputs that overwhelm the victim in computing the output
    - algorithmic complexity attack
- network level:
    - overwhelm victim’s bandwidth/packet processing capacity

denial of service (DoS)

- application level:
  - use inputs that overwhelm the victim in computing the output
  - algorithmic complexity attack
- network level:
  - overwhelm victim’s bandwidth/packet processing capacity
  - DDoS (distributed denial of service)

DoS mitigations

DoS mitigations

- overprovisioning:

DoS mitigations

- overprovisioning:
  - purchase enough physical resources/network bandwidth to avoid DoS

DoS mitigations

- overprovisioning:
  - purchase enough physical resources/network bandwidth to avoid DoS
- packet filters

DoS mitigations

- overprovisioning:
    - purchase enough physical resources/network bandwidth to avoid DoS
- packet filters
    - discard packets where source IP is malicious

DoS mitigations

- overprovisioning:
  - purchase enough physical resources/network bandwidth to avoid DoS
- packet filters
  - discard packets where source IP is malicious
  - examine packet content for malice

DoS mitigations

- overprovisioning:
  - purchase enough physical resources/network bandwidth to avoid DoS
- packet filters
  - discard packets where source IP is malicious
  - examine packet content for malice
  - not very effective against DDoS (many IPs)

firewalls & packet filters

firewalls & packet filters

- firewalls often packet filters—inspect packets

firewalls & packet filters

- firewalls often packet filters—inspect packets
- stateless packet filter

firewalls & packet filters

- firewalls often packet filters—inspect packets
- stateless packet filter
    - no history of previous packets

firewalls & packet filters

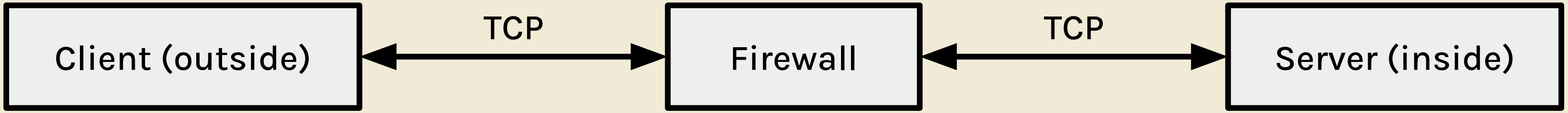
- firewalls often packet filters—inspect packets
- stateless packet filter
  - no history of previous packets
- stateful packet filter

firewalls & packet filters

- firewalls often packet filters—inspect packets
- stateless packet filter
  - no history of previous packets
- stateful packet filter
  - keeps track of history & inbound/outbound connections

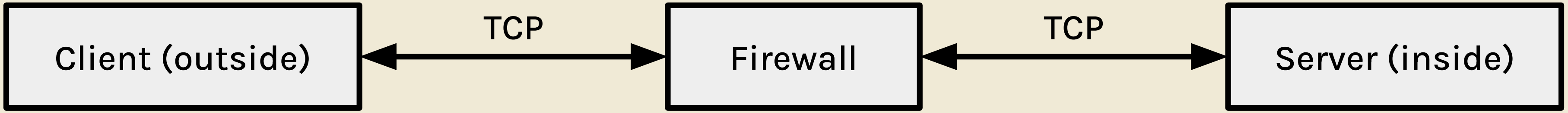
proxy firewalls

- think of the firewall as a (helpful) MITM



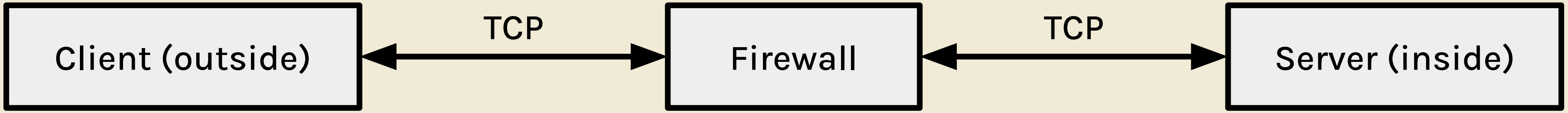
proxy firewalls

- think of the firewall as a (helpful) MITM
- has direct access to the TCP bytestreams



proxy firewalls

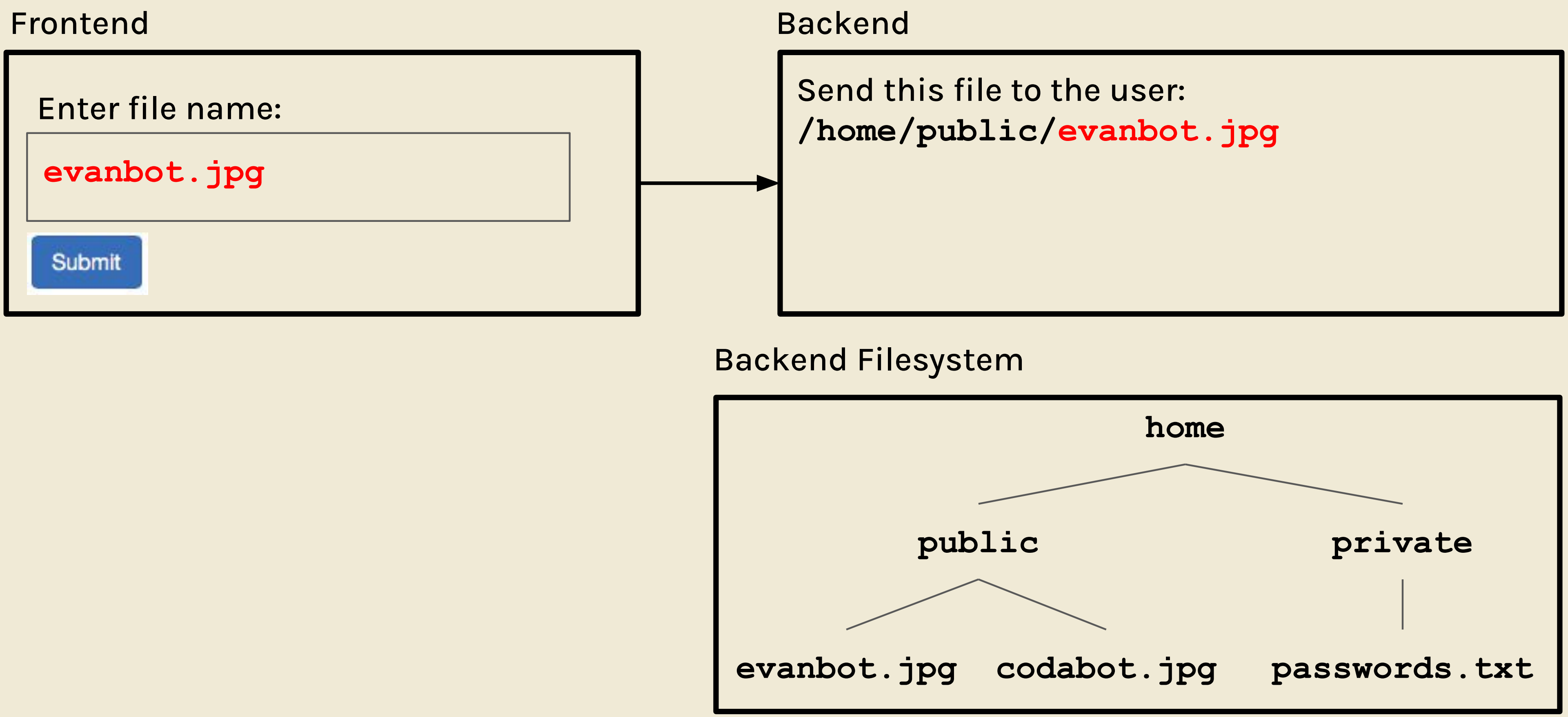
- think of the firewall as a (helpful) MITM
- has direct access to the TCP bytestreams
- proxy can spoof incoming/outgoing IPs



intrusion detection

detect if you can’t prevent

path traversal attack



Frontend

Enter file name:

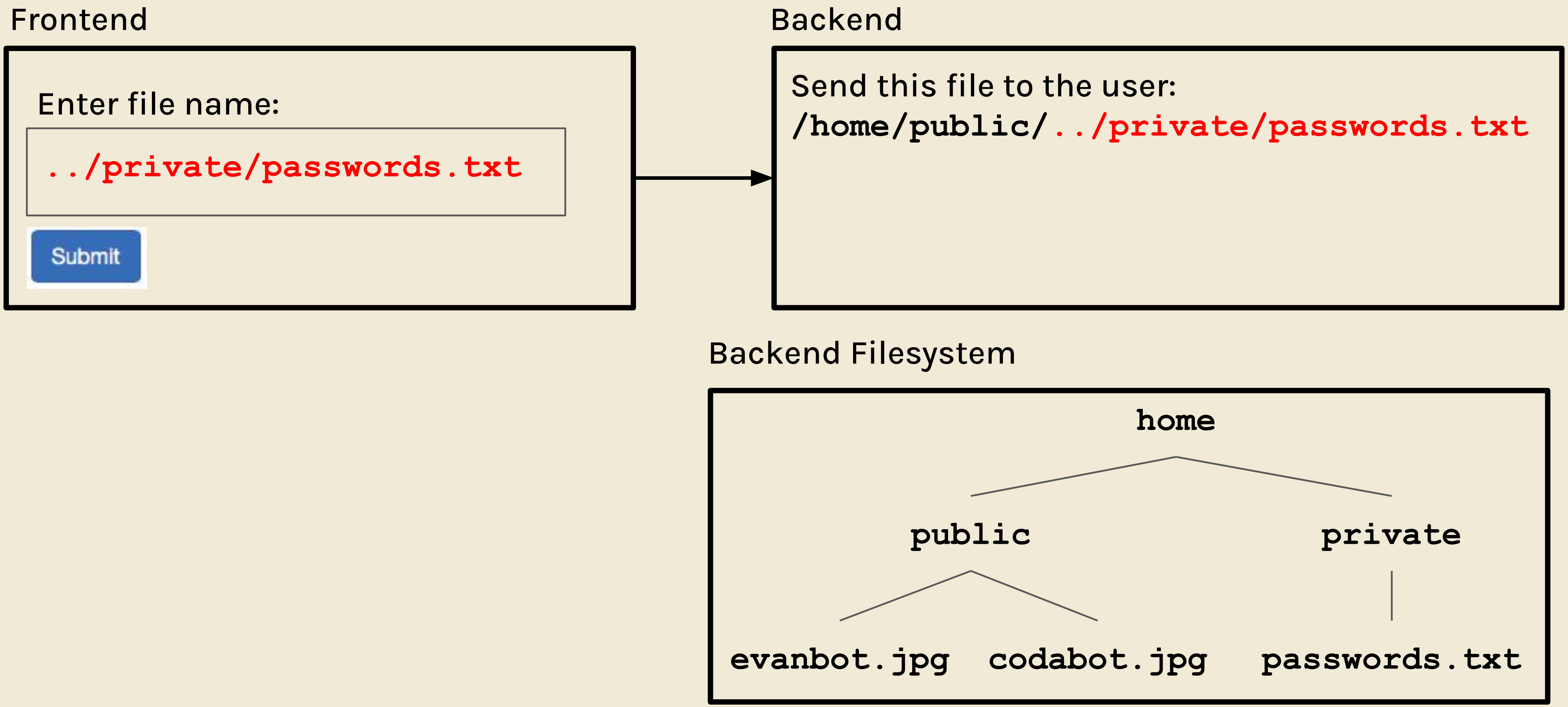
`evanbot.jpg`

Backend

Send this file to the user:
`/home/public/evanbot.jpg`

Backend Filesystem

path traversal attack



Frontend

Enter file name:

`../private/passwords.txt`

Submit

Backend

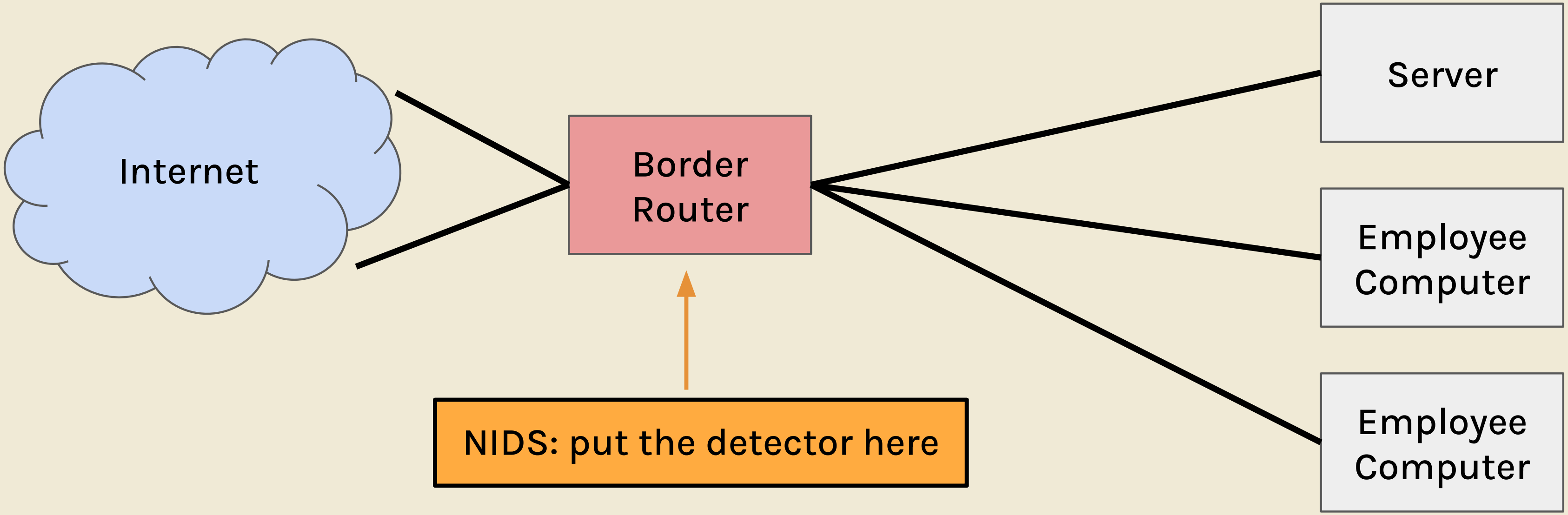
Send this file to the user:

`/home/public/../private/passwords.txt`

Backend Filesystem

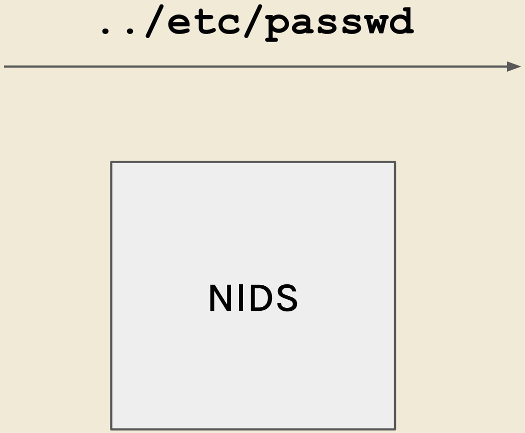
network intrusion detection system

- NIDS: monitors network traffic to detect attacks



NIDS drawback: inconsistency

- NIDS doesn’t know what to do if it sees ambiguous information

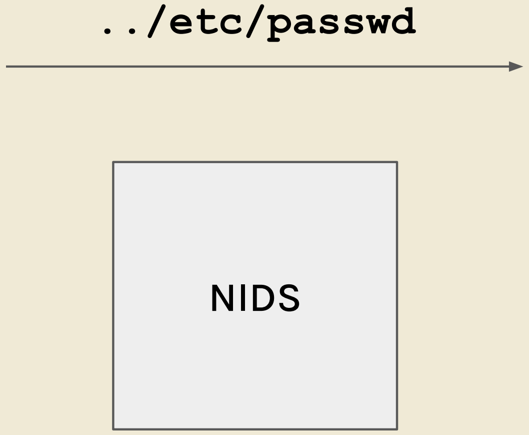


`../etc/passwd`

NIDS

NIDS drawback: inconsistency

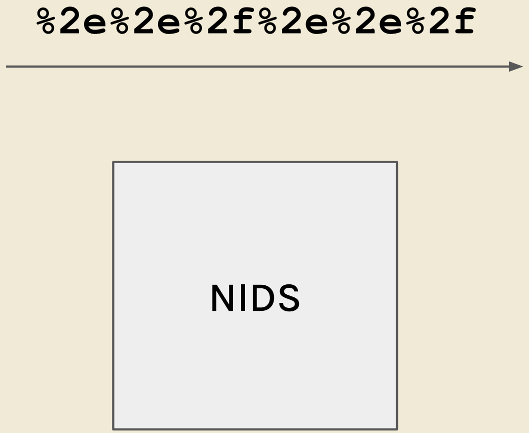
- NIDS doesn’t know what to do if it sees ambiguous information



alert because it looks like path traversal

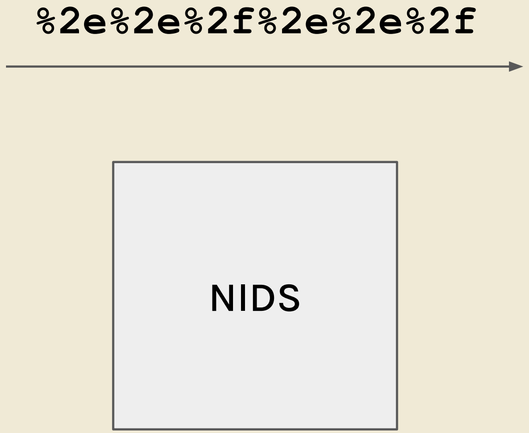
NIDS drawback: inconsistency

- NIDS doesn’t know what to do if it sees ambiguous information



NIDS drawback: inconsistency

- NIDS doesn’t know what to do if it sees ambiguous information



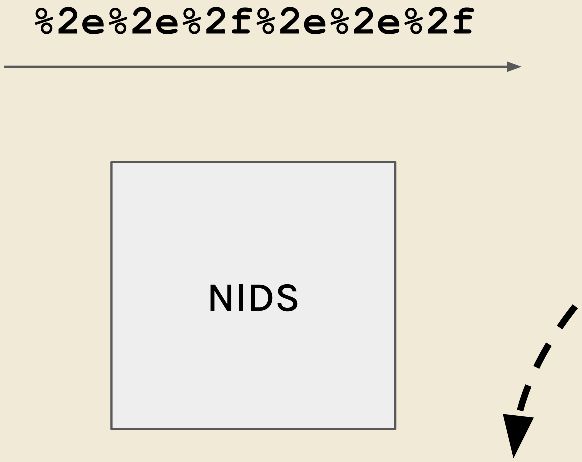
`%2e%2e%2f%2e%2e%2f`

NIDS

don’t alert, doesn’t look like path traversal

NIDS drawback: inconsistency

- NIDS doesn’t know what to do if it sees ambiguous information



wrong, decodes to `../etc/passwd`! after URL percent decoding

don’t alert, doesn’t look like path traversal

NIDS drawbacks

NIDS drawbacks

- **evasion attack:** ambiguous input to bypass NIDS

NIDS drawbacks

- **evasion attack:** ambiguous input to bypass NIDS
  - defenses:

NIDS drawbacks

- **evasion attack:** ambiguous input to bypass NIDS
    - defenses:
        - consider all possible interpretations

NIDS drawbacks

- **evasion attack:** ambiguous input to bypass NIDS
  - defenses:
    - consider all possible interpretations
    - enforce normalized form for input

NIDS drawbacks

- **evasion attack:** ambiguous input to bypass NIDS
  - defenses:
    - consider all possible interpretations
    - enforce normalized form for input
    - flag possible evasions

NIDS drawbacks

- **evasion attack**: ambiguous input to bypass NIDS
  - defenses:
    - consider all possible interpretations
    - enforce normalized form for input
    - flag possible evasions
- encrypted traffic (TLS)

NIDS drawbacks

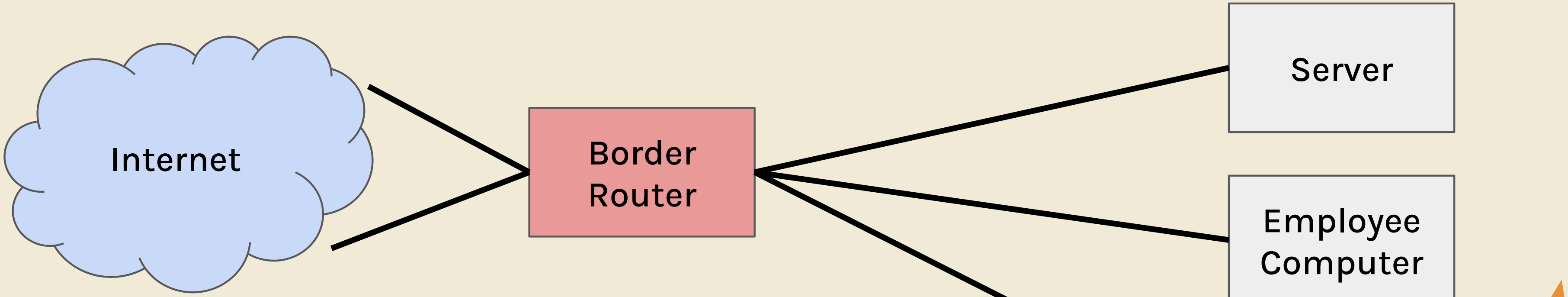
- **evasion attack:** ambiguous input to bypass NIDS
  - defenses:
    - consider all possible interpretations
    - enforce normalized form for input
    - flag possible evasions
- encrypted traffic (TLS)
  - can just give NIDS all private keys on network

NIDS drawbacks

- **evasion attack**: ambiguous input to bypass NIDS
    - defenses:
        - consider all possible interpretations
        - enforce normalized form for input
        - flag possible evasions
- encrypted traffic (TLS)
    - can just give NIDS all private keys on network
        - why is this unideal?

host-based intrusion detection system

- HIDS: detector installed on each end system



HIDS: put the detector here

HIDS benefits/drawbacks

HIDS benefits/drawbacks

- benefits:

HIDS benefits/drawbacks

- benefits:
    - less evasion, interpret packets as host does

HIDS benefits/drawbacks

- benefits:
    - less evasion, interpret packets as host does
    - works for encrypted messages

HIDS benefits/drawbacks

- benefits:
  - less evasion, interpret packets as host does
  - works for encrypted messages
  - performance scales better

HIDS benefits/drawbacks

- benefits:
    - less evasion, interpret packets as host does
    - works for encrypted messages
    - performance scales better
    - prevent against in-network attacker

HIDS benefits/drawbacks

- benefits:
  - less evasion, interpret packets as host does
  - works for encrypted messages
  - performance scales better
  - prevent against in-network attacker
- drawbacks:

HIDS benefits/drawbacks

- benefits:
    - less evasion, interpret packets as host does
    - works for encrypted messages
    - performance scales better
    - prevent against in-network attacker
- drawbacks:
    - expensive—one detector per end host

HIDS benefits/drawbacks

- benefits:
  - less evasion, interpret packets as host does
  - works for encrypted messages
  - performance scales better
  - prevent against in-network attacker
- drawbacks:
  - expensive—one detector per end host
  - evasion attacks still sometimes possible

logging

- analyze log files generated by end systems
  - likely overnight, less traffic
- detect **after** attacks have happened
- very cheap

styles of detection

styles of detection

- **signature-based:** flag activity matching known attack
    - blacklisting (path traversal, known shellcode)

styles of detection

- **signature-based**: flag activity matching known attack
  - blacklisting (path traversal, known shellcode)
- **specification-based**: flag disallowed behavior
  - whitelisting (specify allowed behavior)

styles of detection

- **signature-based**: flag activity matching known attack
    - blacklisting (path traversal, known shellcode)
- **specification-based**: flag disallowed behavior
    - whitelisting (specify allowed behavior)
- **anomaly-based:** develop model for “usual” behavior and flag if deviating

styles of detection

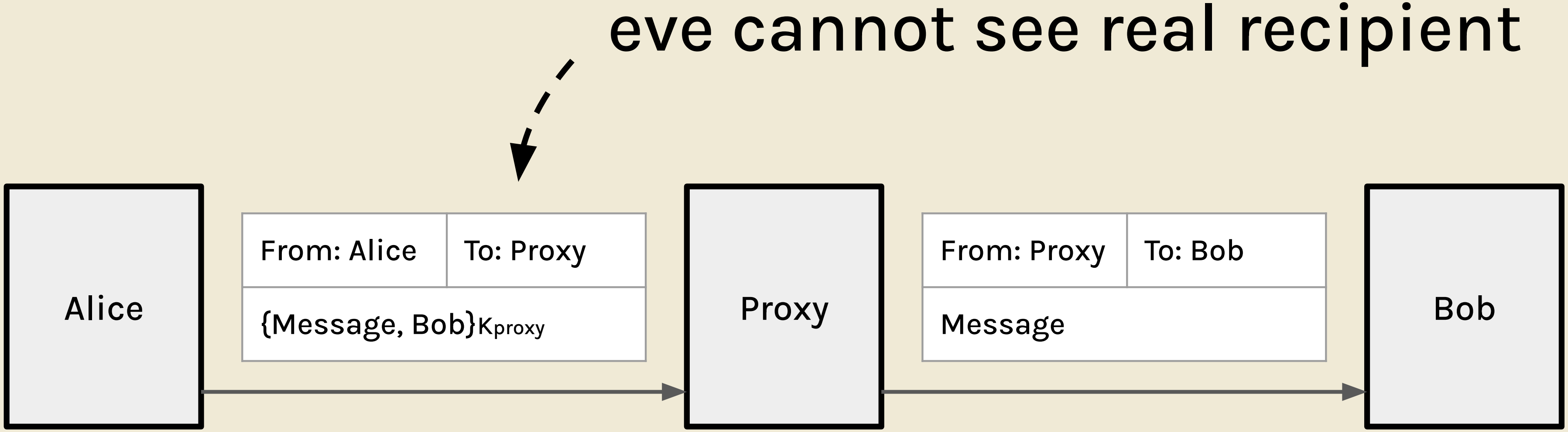
- **signature-based**: flag activity matching known attack
  - blacklisting (path traversal, known shellcode)
- **specification-based**: flag disallowed behavior
  - whitelisting (specify allowed behavior)
- **anomaly-based:** develop model for “usual” behavior and flag if deviating
- **behavioral:** look for evidence of compromise
  - look at ~~input~~ *actions* triggered by input

worksheet

(on 161 website)

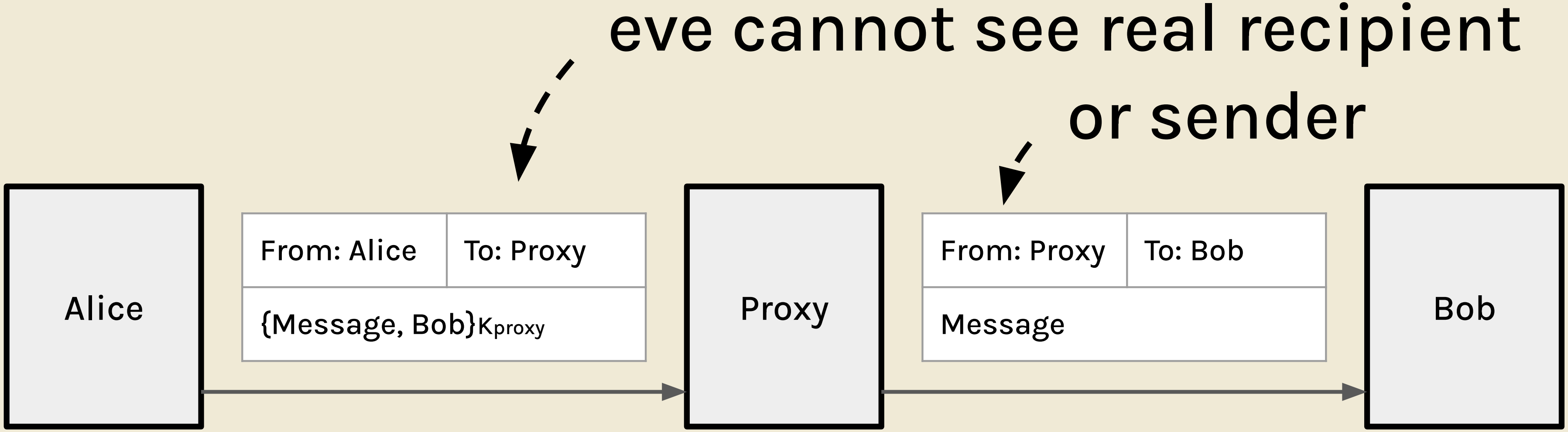
anonymity, tor

proxy recap



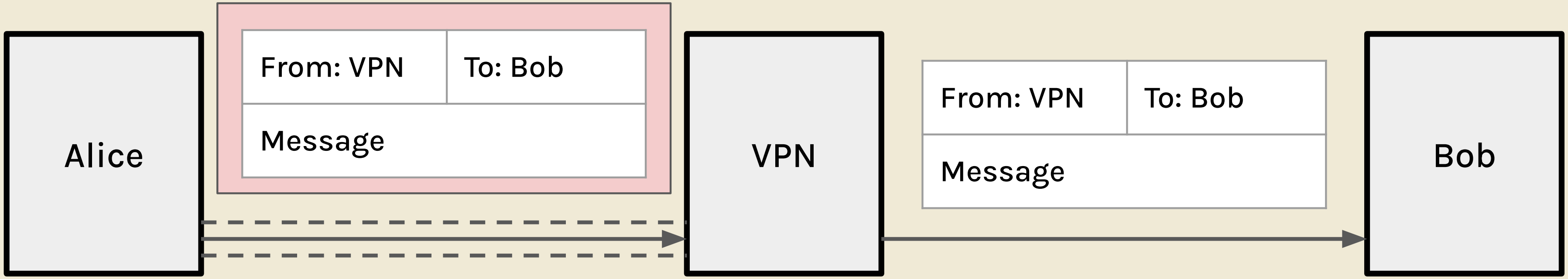
- proxy acts as MITM—rests on **application layer**

proxy recap



- proxy acts as MITM—rests on **application layer**

VPN



- access an internal network via a VPN
- send data as if from a different network

issues with proxies/VPNs

issues with proxies/VPNs

- performance: additional hops around Internet

issues with proxies/VPNs

- performance: additional hops around Internet
- VPNs cost money

issues with proxies/VPNs

- performance: additional hops around Internet
- VPNs cost money
- trusting the proxy (can see sender and recipient's identities, have to trust it won't be given out)

Tor (the onion router)

Tor (the onion router)

- idea: use multiple proxies to hide source/dest
    - call these **relays**

Tor (the onion router)

- idea: use multiple proxies to hide source/dest
    - call these **relays**
- **Tor network:** a network of many relays

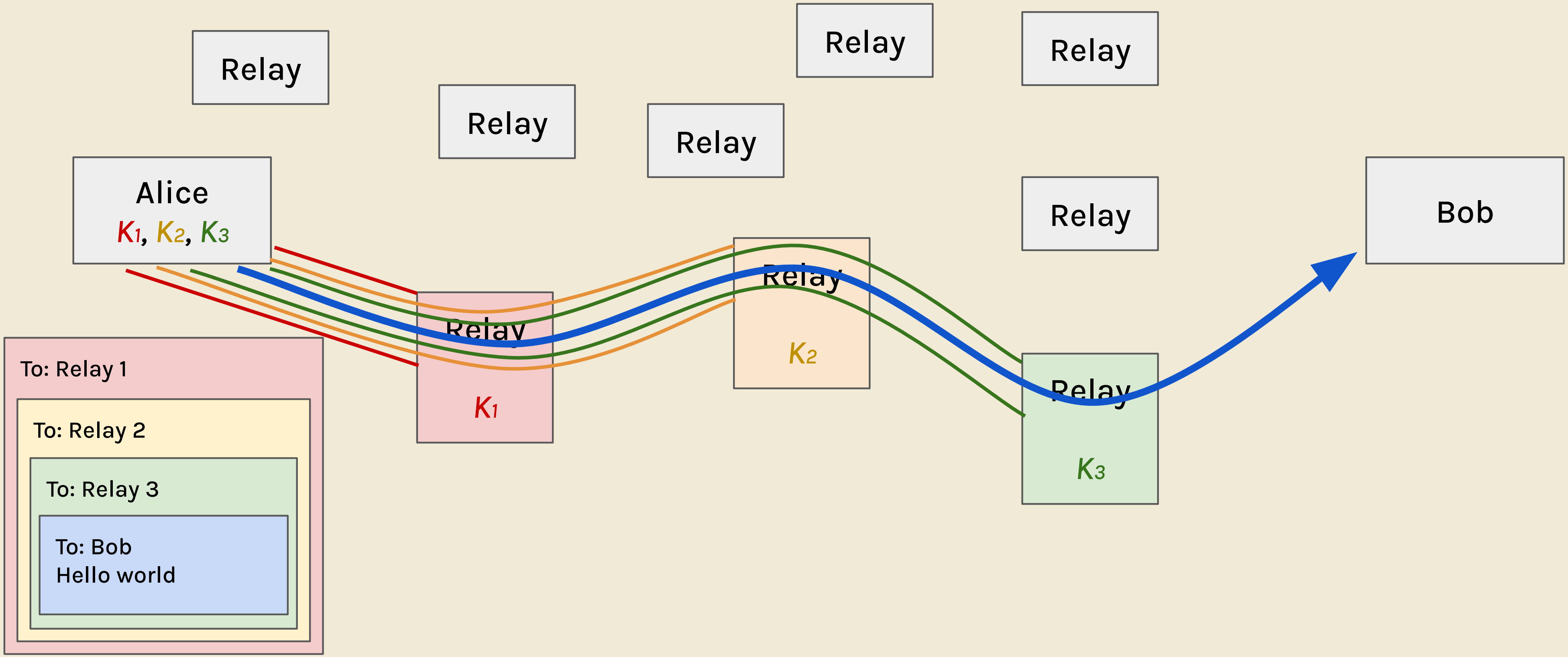
Tor (the onion router)

- idea: use multiple proxies to hide source/dest
  - call these **relays**
- **Tor network:** a network of many relays
- **directory server:** lists all Tor relays + public keys

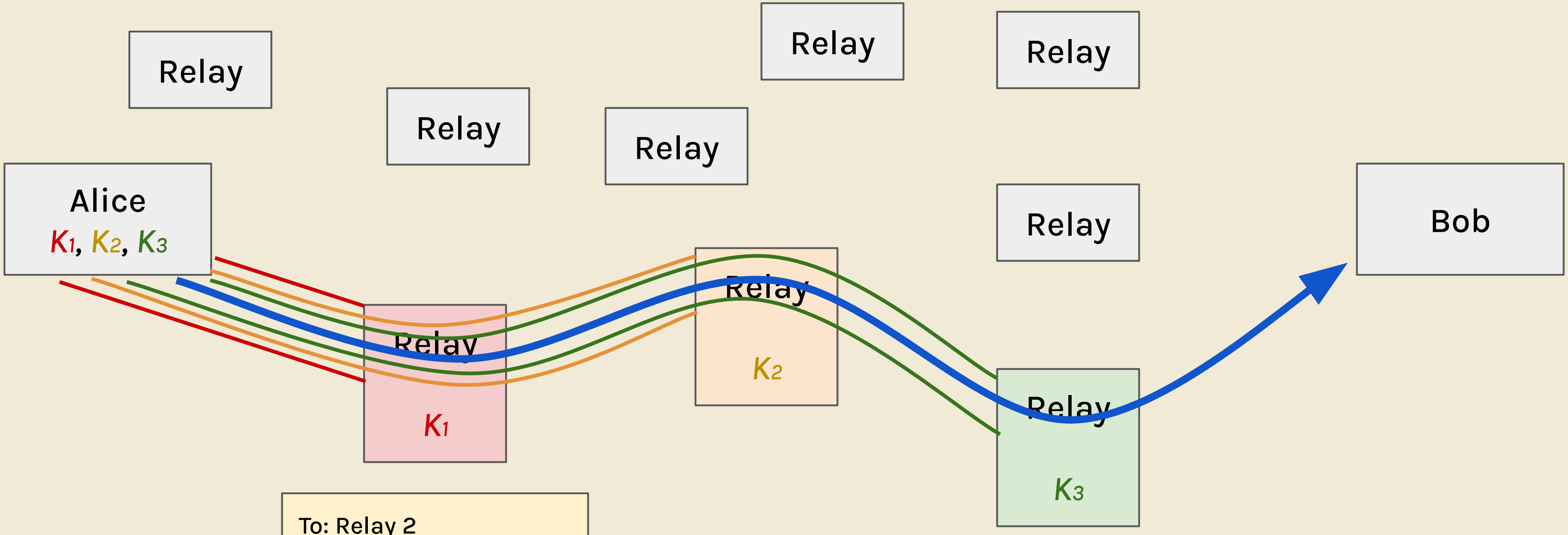
Tor (the onion router)

- idea: use multiple proxies to hide source/dest
  - call these **relays**
- **Tor network:** a network of many relays
- **directory server:** lists all Tor relays + public keys
- **Tor browser:** browser based on Firefox

example Tor circuit



example Tor circuit

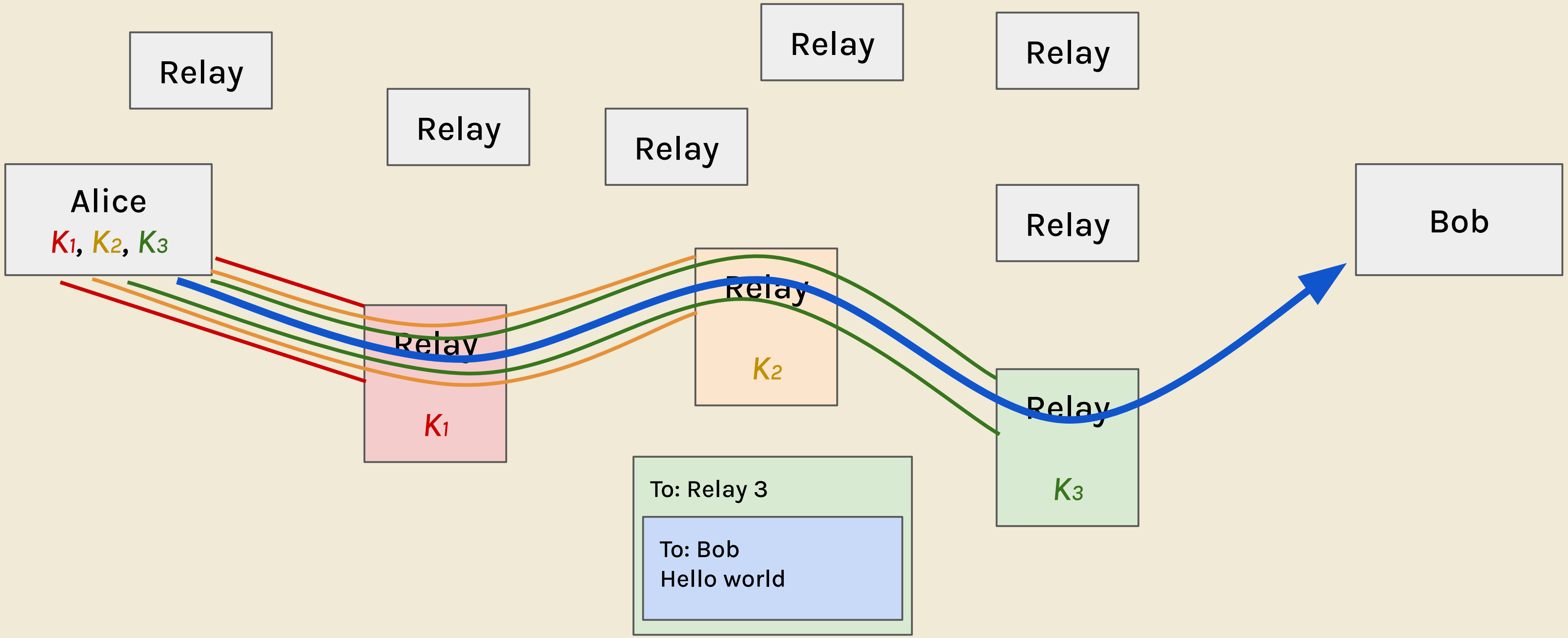


To: Relay 2

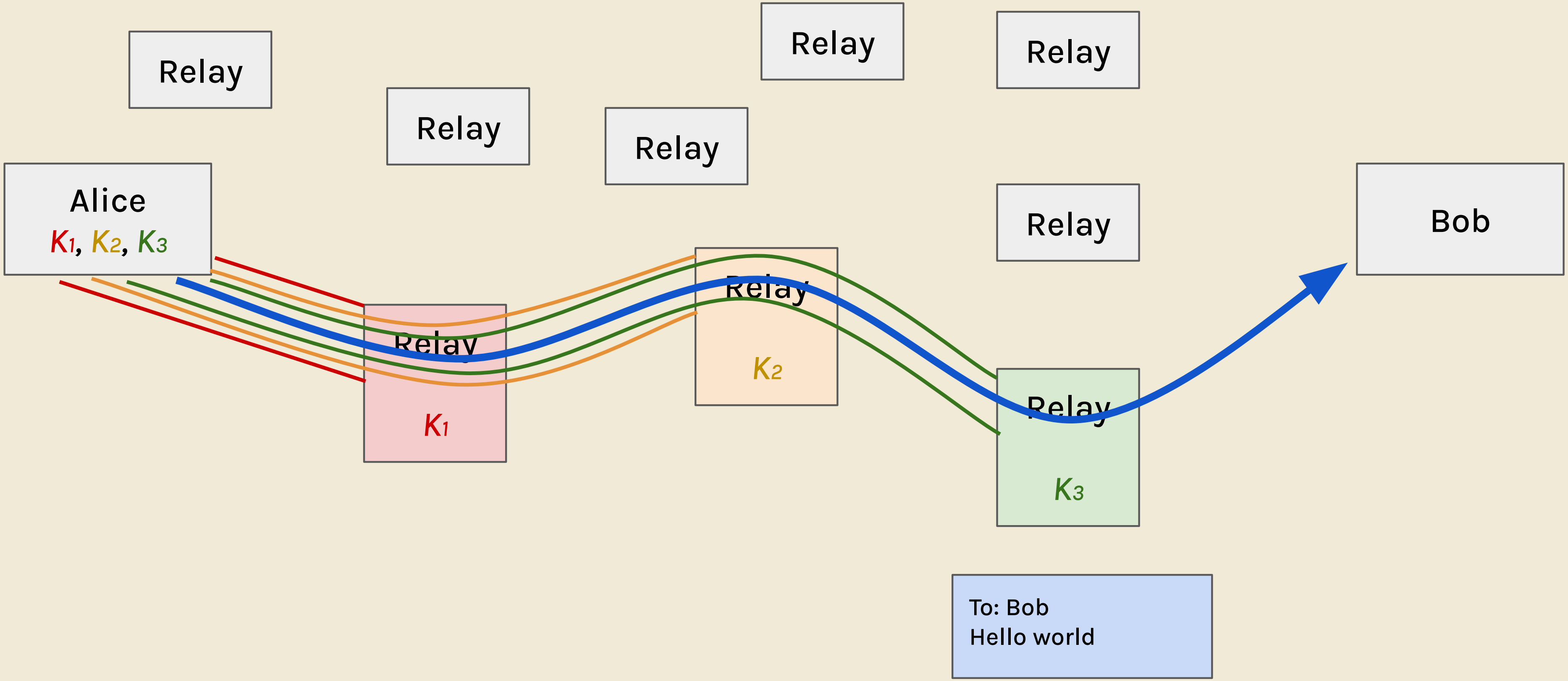
To: Relay 3

To: Bob
Hello world

example Tor circuit

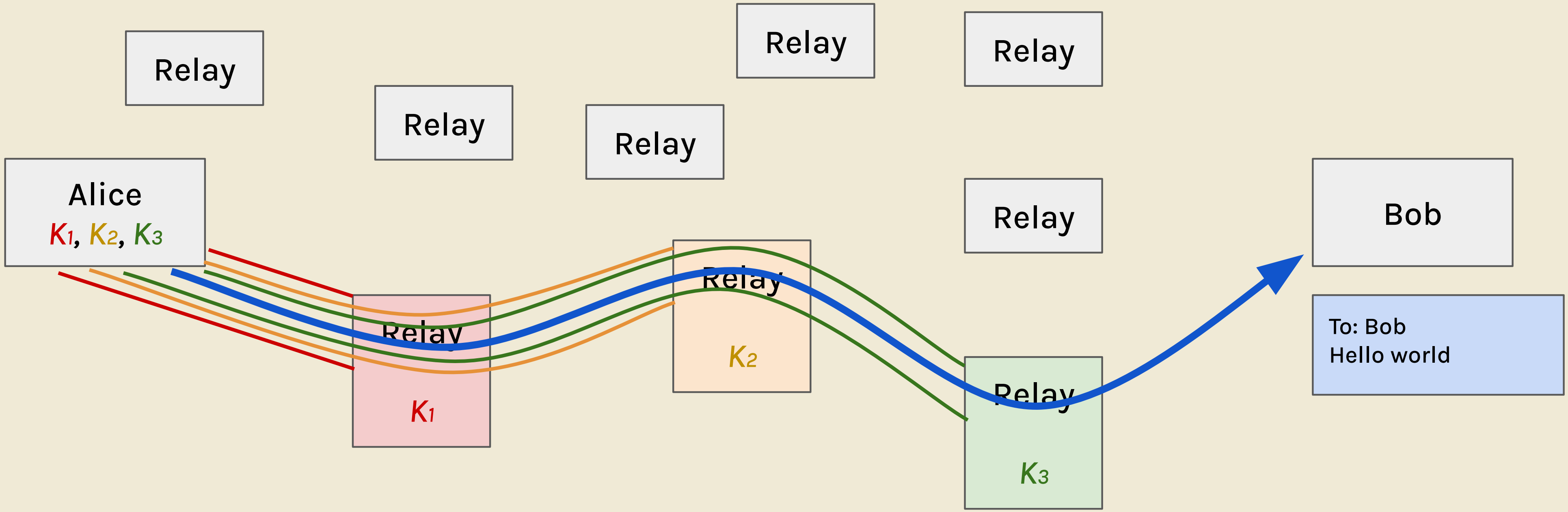


example Tor circuit



To: Bob
Hello world

example Tor circuit



Tor weaknesses

Tor weaknesses

- doesn’t defend against **global adversaries**
    - can observe timing of alice sending + bob receiving messages and link them

Tor weaknesses

- doesn’t defend against **global adversaries**
  - can observe timing of alice sending + bob receiving messages and link them
- **collusion:** if all nodes work together and share information

Tor weaknesses

- doesn’t defend against **global adversaries**
  - can observe timing of alice sending + bob receiving messages and link them
- **collusion:** if all nodes work together and share information
  - if (at least) one node is honest, anonymous

Tor weaknesses

- doesn’t defend against **global adversaries**
  - can observe timing of alice sending + bob receiving messages and link them
- **collusion:** if all nodes work together and share information
  - if (at least) one node is honest, anonymous
  - defense: reputable “guard” node = entry node

Tor weaknesses

- doesn’t defend against **global adversaries**
  - can observe timing of alice sending + bob receiving messages and link them
- **collusion:** if all nodes work together and share information
  - if (at least) one node is honest, anonymous
  - defense: reputable “guard” node = entry node
- **distinguishable**: can tell when user is using Tor

Tor weaknesses

- doesn’t defend against **global adversaries**
  - can observe timing of alice sending + bob receiving messages and link them
- **collusion:** if all nodes work together and share information
  - if (at least) one node is honest, anonymous
  - defense: reputable “guard” node = entry node
- **distinguishable**: can tell when user is using Tor
  - use Tor bridges: non-public entry node

Tor onion services

Tor onion services

- websites that are only accessible through Tor

Tor onion services

- websites that are only accessible through Tor
    - want to have anon. server locations

Tor onion services

- websites that are only accessible through Tor
  - want to have anon. server locations
- sometimes called the dark web

Tor onion services

- websites that are only accessible through Tor
    - want to have anon. server locations
- sometimes called the dark web
- sparknotes version (out of scope):
    - client sends **rendezvous point** location to server-published **introduction point** via Tor circuit, which sends the rendezvous point to the server

Tor onion services

- websites that are only accessible through Tor
    - want to have anon. server locations
- sometimes called the dark web
- sparknotes version (out of scope):
    - client sends **rendezvous point** location to server-published **introduction point** via Tor circuit, which sends the rendezvous point to the server
    - communicate through rendezvous point

hack of the day v2

hack of the day v2

- over 25% of Tor exit relays spied on users
  - threat actor controlled >27% of exit relays

hack of the day v2

- over 25% of Tor exit relays spied on users
    - threat actor controlled >27% of exit relays
    - Tor dropped these from Tor network

hack of the day v2

- over 25% of Tor exit relays spied on users
  - threat actor controlled >27% of exit relays
  - Tor dropped these from Tor network
  - hacker used SSL stripping to replace bitcoin addresses and redirect transactions to their wallets

hack of the day v2

- over 25% of Tor exit relays spied on users
    - threat actor controlled >27% of exit relays
    - Tor dropped these from Tor network
    - hacker used SSL stripping to replace bitcoin addresses and redirect transactions to their wallets
    - Tor Project says to websites—enable HTTPS by default, deploy .onion sites

worksheet

(on 161 website)

feedback

bit.ly/abhifeedback

slides: **bit.ly/cs161-disc**