CSCI-351
Data communication and Networks

Lecture 15: Security Basics

Something that you do not want to memorize

The slide is built with the help of Prof. Alan Mislove, Christo Wilson, and David Choffnes's class

2 Outline

- Principles
- Basics
- Vulnerabilities

3 principles of information security

CIA Triangles:

- Confidentiality
- Integrity
- Availability

Confidentiality

"Hey, we're attacking at dawn!"

- Data must only be released to *authorized principals*
- Cryptography has historically focused on providing confidentiality
  - But, there are other mechanisms
- Can have a temporal aspect

Integrity

"Retreat at dawn."

- Data must not be modified (in an undetectable manner)
- What constitutes a modification?
  - Corruption
  - Dropped, replayed, or reordered messages
- Cryptography has also historically provided this
  - e.g, (cryptographic) hash functions, HMAC

Availability

“Xfk3^#M3mf a ___ q3rf” – jamming results in garbled message

- Data and resources must be accessible when required
- Related to integrity, but more concerned with denial of service (DoS) attacks
  - Resource exhaustion (e.g., CPU, memory, network bandwidth)
  - Usually easy to perform, can be difficult to defend

Authenticity

Enemy commander: "Attack at dawn."

- Establishment of identity
  - Or, verification of "genuineness"
- Again, cryptography has long considered this
  - e.g., HMAC, signatures

Non-repudiation

"I never said to attack at dawn!"

- Data must be bound to identity
  - Prevents denial of message transmission or receipt
- Cryptographic techniques
  - e.g., HMAC, certificates

Access Control

- Policy specifying how entities can interact with resources
  - i.e., *Who* can access *what*?
  - Requires authentication and authorization

Authentication

- Verification of identity claim made by a subject on behalf of a principal
- Involves examination of *factors*, or *credentials*
  - Something you *have* – e.g., a badge
  - Something you *know* – e.g., a password
  - Something you *are* – e.g., your fingerprint
- Desirable properties include being *unforgeable*, *unguessable*, and *revocable*

Outline

- Principals
- Basics
- Vulnerabilities

Security Principles

- We've seen some basic properties, policies, mechanisms, models, and approaches to security
- But, designing secure systems (and breaking them) remains an art
- Security principles help bridge the gap between art and science
  - Let's look at a few

Separation of Privilege

Privilege, or authority, should only be distributed to subjects that require it

- Some components of a system should be less privileged than others
  - Not every subject needs the ability to do everything
  - Not every subject is deserving of full trust
- Contrast with "ambient authority"

Least Privilege

Subjects should possess only that authority that is required to operate successfully

- Closely related to separation of privilege
  - Not only should privilege be separated, but subjects should have the *least* amount necessary to perform a task

Security vs. Usability

- Security often comes with a trade-off between the level of protection provided and ease-of-use
  - Systems that try to provide very strong security guarantees tend to be unusable in practice
  - Completely insecure systems are usually easy to use

Outline

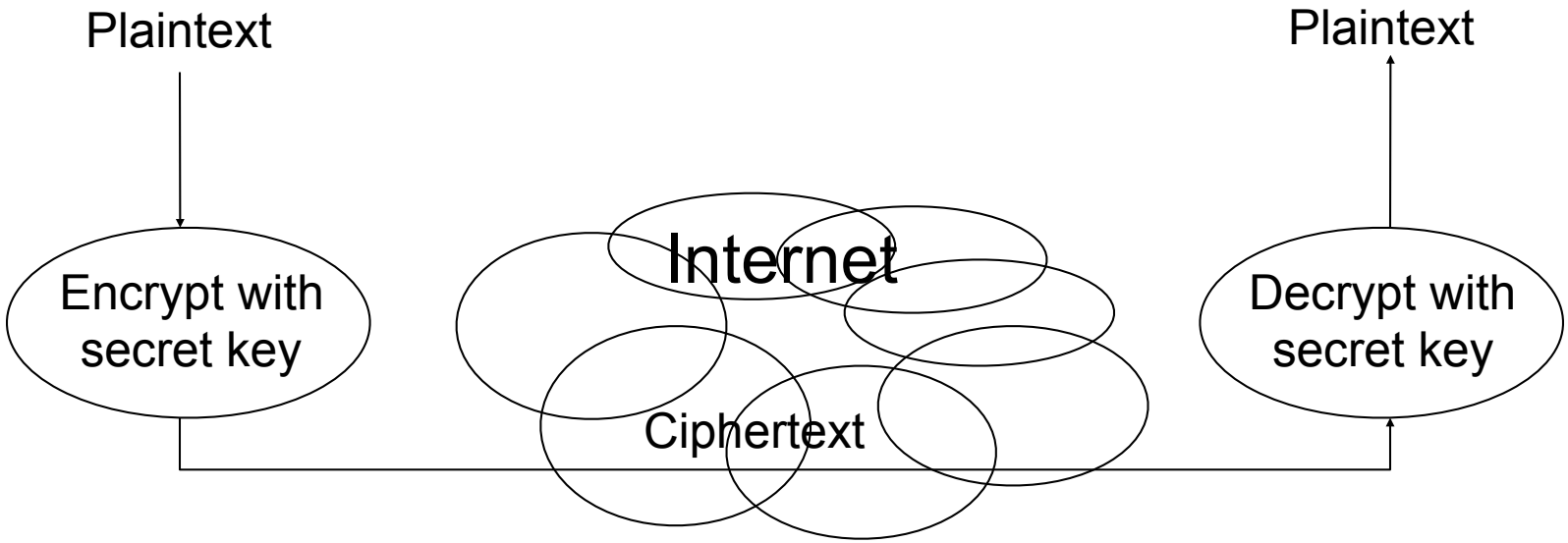
- Principals
- Basics
- Vulnerabilities

Cryptographic Algorithms

- Security foundation: cryptographic algorithms
  - Secret key cryptography, e.g. Data Encryption Standard (DES)
  - Public key cryptography, e.g. RSA algorithm
  - Message digest, e.g. MD5

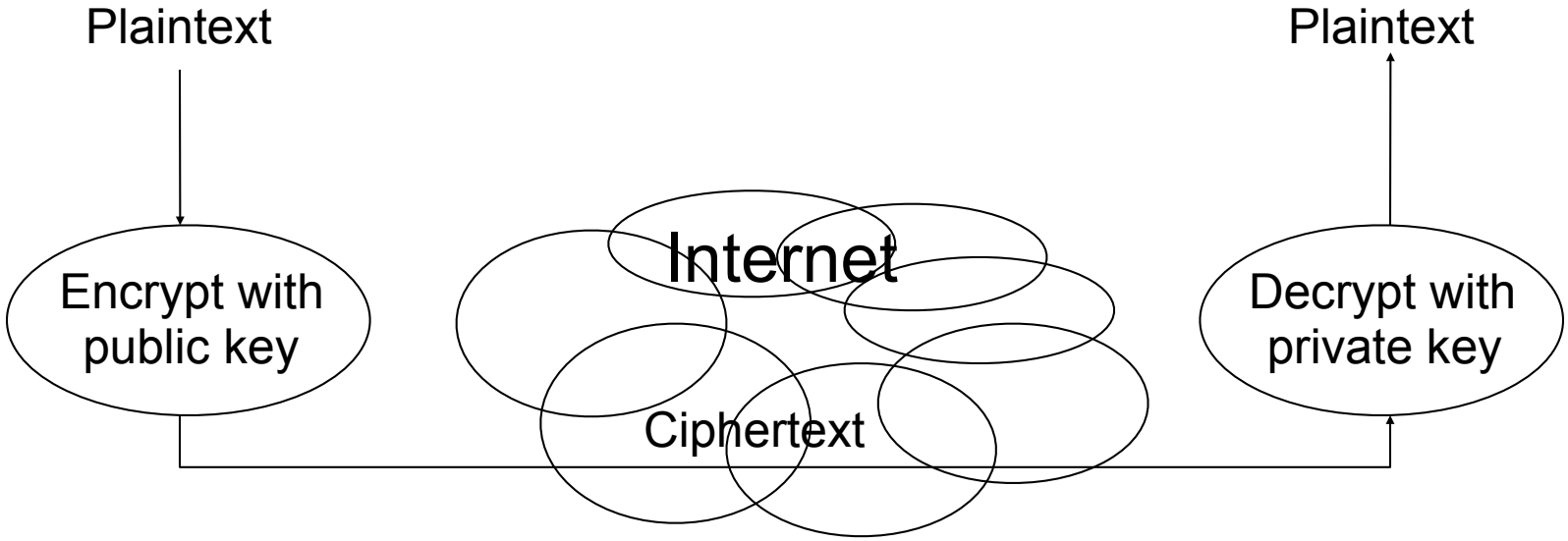
Symmetric Key

- Both the sender and the receiver use the same secret keys



Public-Key Cryptography: RSA

- Sender uses a public key
  - Advertised to everyone
- Receiver uses a private key

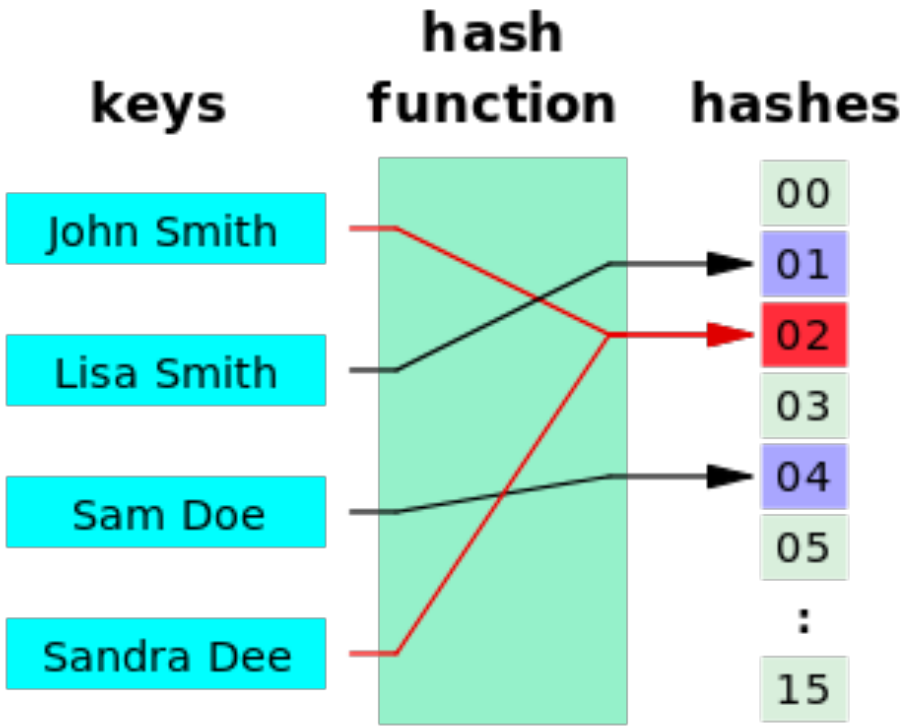


- See RSA if you're interested in

Hash

20

- Time complexity
- Obtaining a hash value is O(1)
- Conjecturing keys from the hash is….
  - In case of sha256, it takes 10**57 minutes (theoretically)

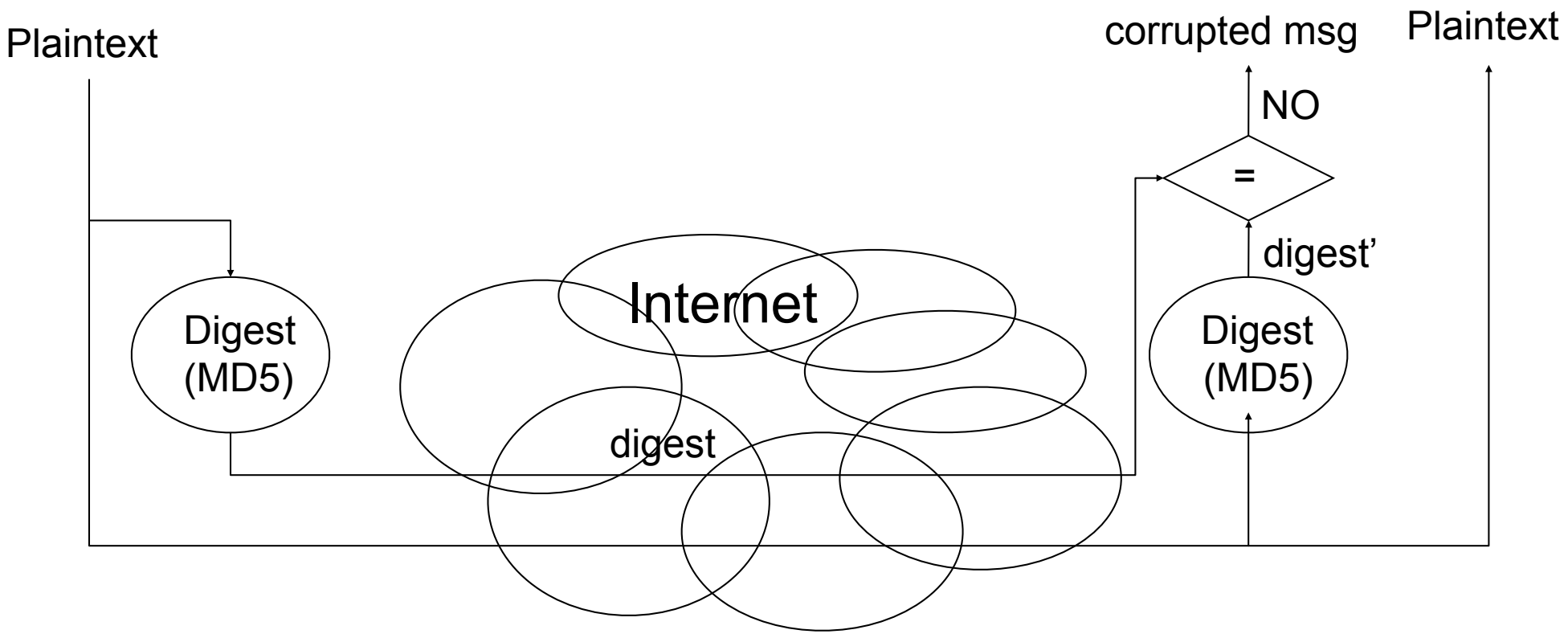


Message Digest (MD) MD5/SHA1

- Can provide data integrity
  - Used to verify the authenticity of a message
- Idea: compute a hash value on the message and send it along with the message
- Receiver can apply the same hash function on the message and see whether the result coincides with the received hash
- Very hard to forge a message that produces the same hash value
  - i.e. Message -> hash is easy
  - Hash -> Message is hard
  - Compare to other error detection methods (CRC, parity, etc)

MD 5 (cont'd)

- Basic property: digest operation very hard to invert
  - Send the digest via a different channel



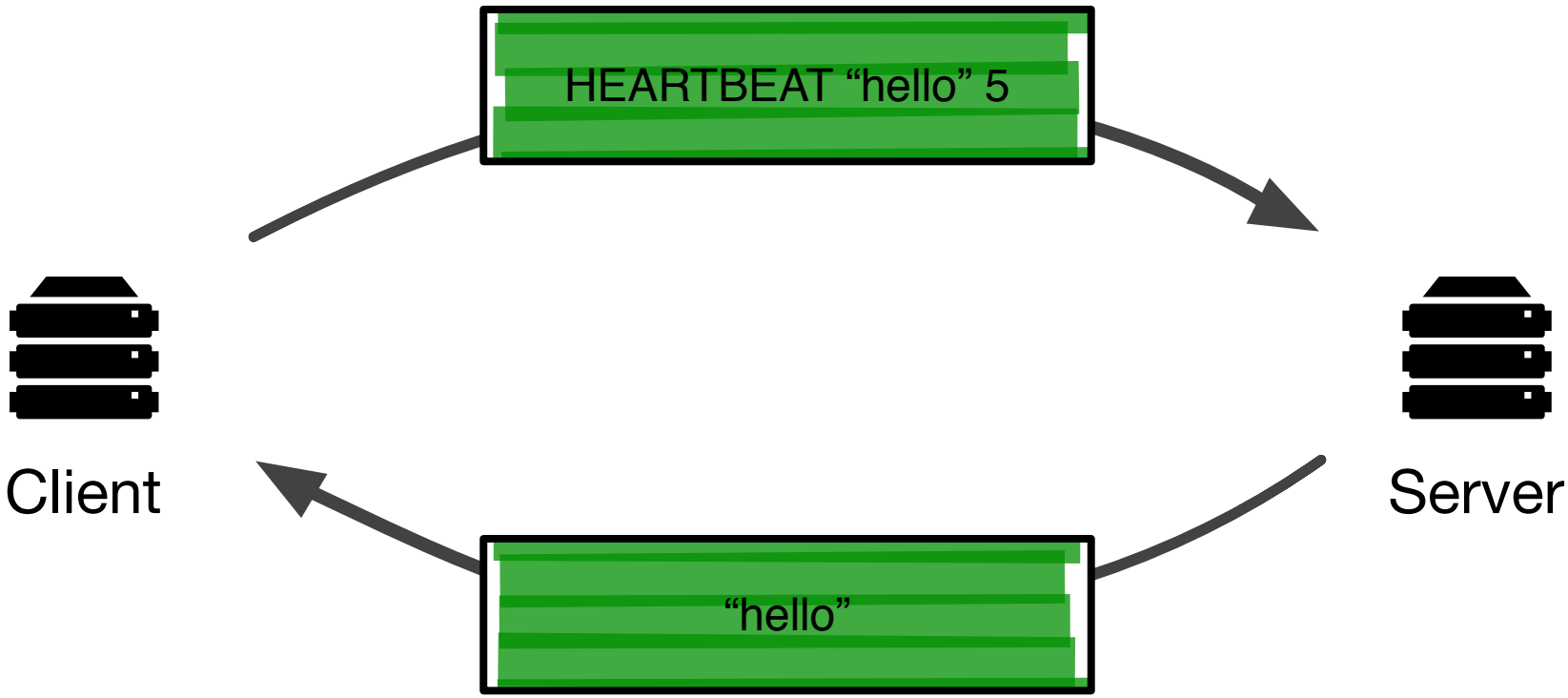
Outline

- Principals
- Basics
- An example of Vulnerabilities

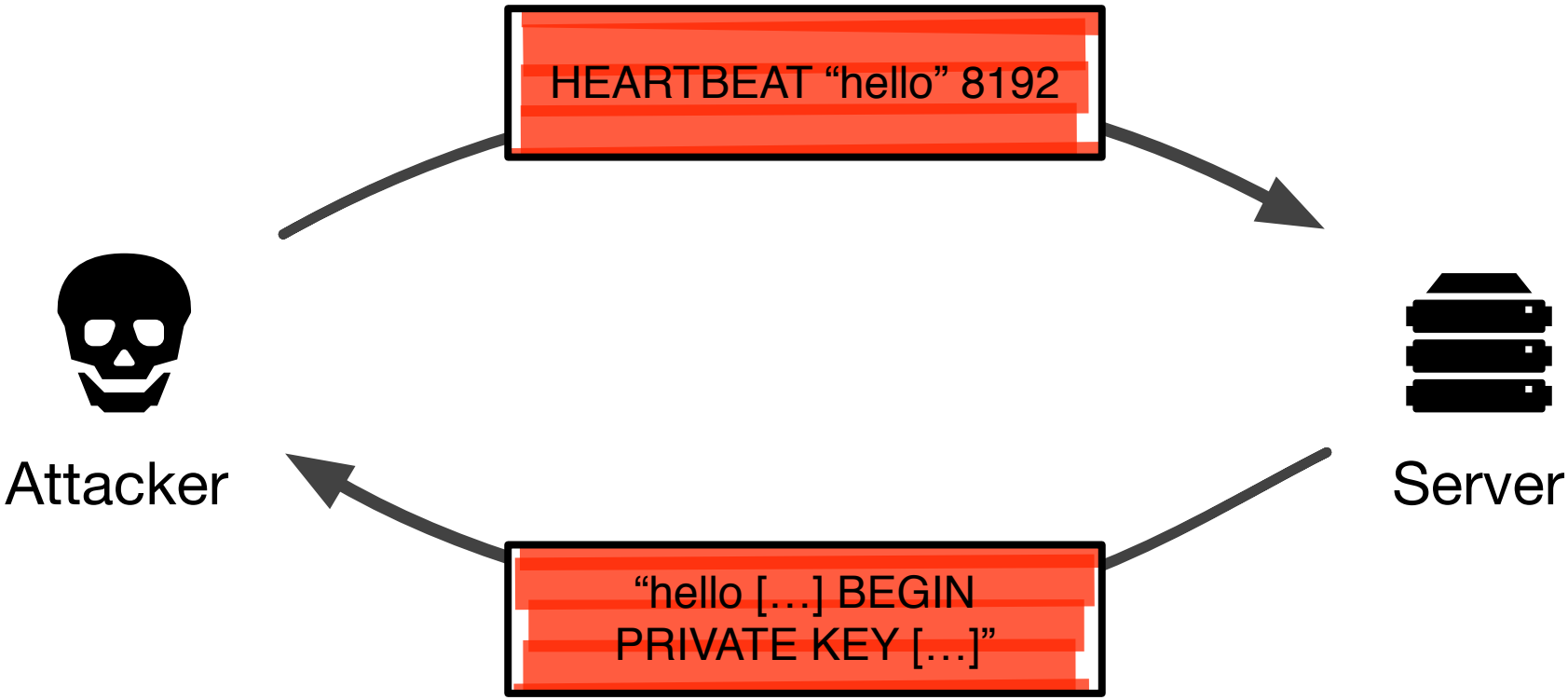
Heartbleed

- Serious vulnerability discovered in OpenSSL in April 2014
  - Involves a bug in the TLS heartbeat extension
- Allows adversaries to read memory of vulnerable services
  - i.e., buffer over-read vulnerability
  - Discloses addresses, sensitive data, potentially TLS secret keys
- Major impact
  - OpenSSL is the de facto standard implementation of TLS, so used everywhere
  - Many exposed services, often on difficult-to-patch devices
  - *Trivial* to exploit

Heartbleed



Heartbleed


HEARTBEAT “hello” 8192
Attacker
Server
“hello […] BEGIN
PRIVATE KEY […]”

THE ACCIDENTAL LEAK —

Google goes down after major BGP mishap routes traffic through China

Google says it doesn't believe leak was malicious despite suspicious appearances.

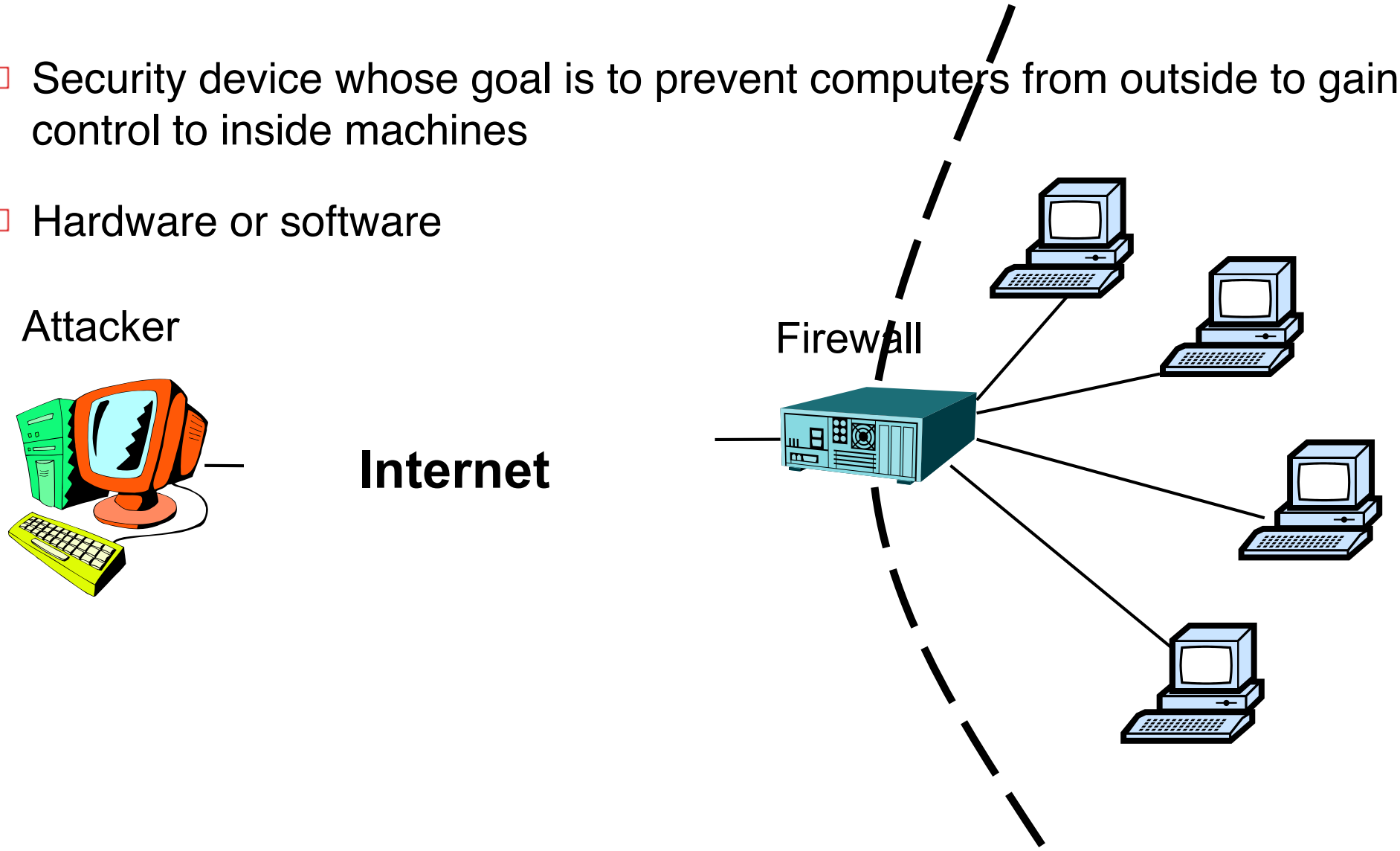
DAN GOODIN - 11/13/2018, 2:25 AM

https://www.flickr.com/photos/bjtsnadow/5668460525

Enlarge

Firewall

- Security device whose goal is to prevent computers from outside to gain control to inside machines
- Hardware or software



Firewall (cont'd)

- Restrict traffic between Internet and devices (machines) behind it based on
  - Source address and port number
  - Payload
  - Stateful analysis of data
- Examples of rules
  - Block any external packets not for port 80
  - Block any email with an attachment
  - Block any external packets with an internal IP address
    - Ingress filtering

Firewalls: Properties

- Easier to deploy firewall than secure all internal hosts
- Doesn't prevent user exploitation
- Tradeoff between availability of services (firewall passes more ports on more machines) and security
  - If firewall is too restrictive, users will find way around it, thus compromising security
  - E.g., have all services use port 80
- Can't prevent problem from spreading from within