

# Security Topics in Networking and Distributed Systems

## CS 590D

### **Lecture 13: DNS: Attacks and Countermeasures**

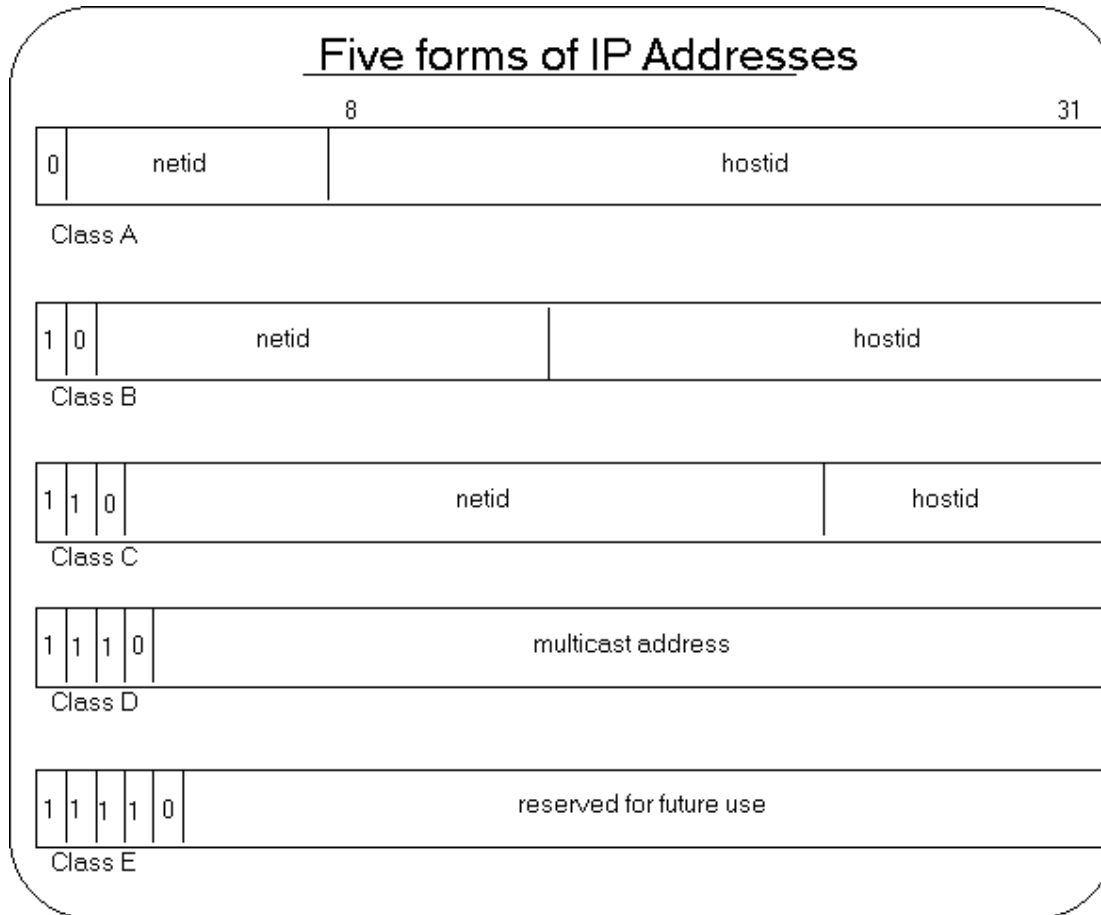
Department of Computer Sciences  
Purdue University

# References

- Threat Analysis Of The Domain Name System. D. Atkins and R. Austein.
- Public Key Validation for the DNS Security Extensions Daniel Massey, Ed Lewis, Olafur Gudmundsson, Russ Mundy and Allison Mankin. DISCEX 2003.
- A new approach to DNS security (DNSSEC) Giuseppe Ateniese and Stefan Mangard, CCS 2001.



# IP address: a reminder



Class A:  
1.0.0.0 - 126.0.0.0

Class B:  
127.0.0.0 - 191.255.0.0

Class C:  
192.0.0.0 - 223.255.255.0

Class D:  
224.0.0.0 - 240.0.0.0

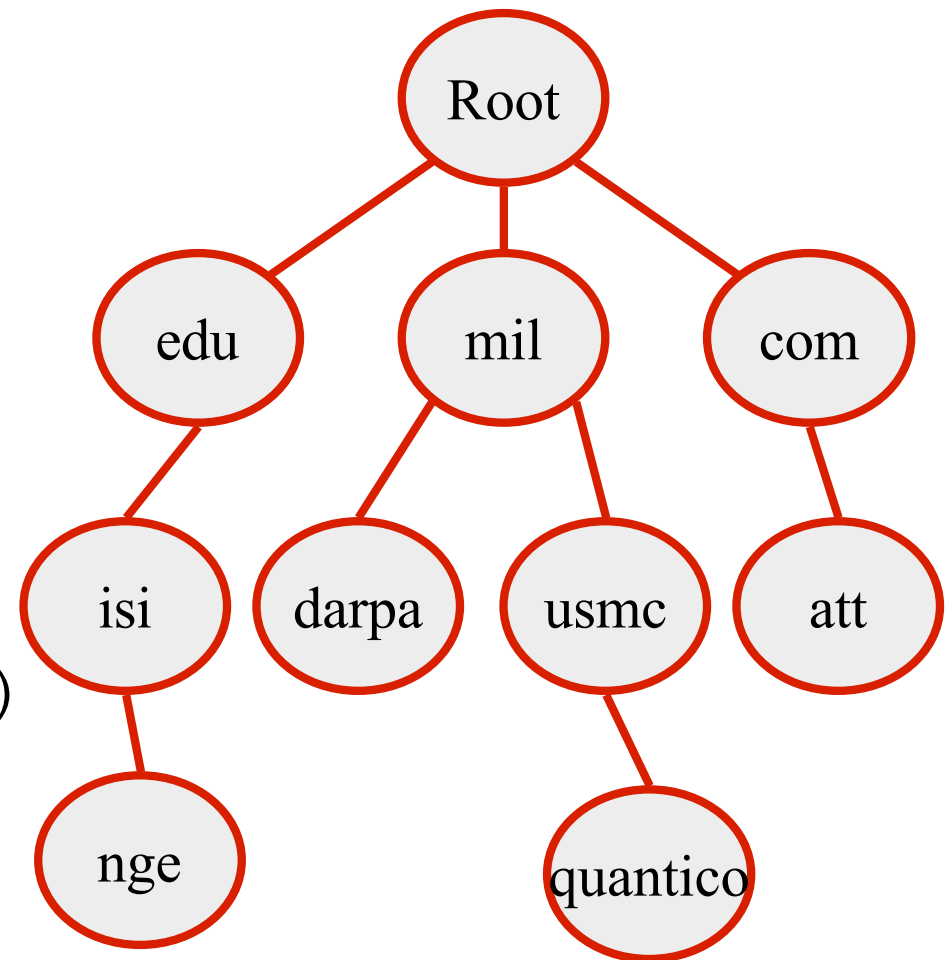
Class E:  
241.0.0.0 - 248.0.0.0

# Domain Name System (DNS)

- People prefer names to identify computers instead of numbers
- DNS: Distributed, hierarchical database that maps host names with IP addresses
- Almost any application uses DNS
- If DNS is not working many applications will be crippled
- Uses UDP

# DNS (cont.)

- Tree structure
  - Divided into zones
  - Delegating responsibilities
- ICANN oversees the domain name assignments
- Name servers
  - Authoritative information (hints to whom might be able to answer the request)
  - Cached data updated periodically



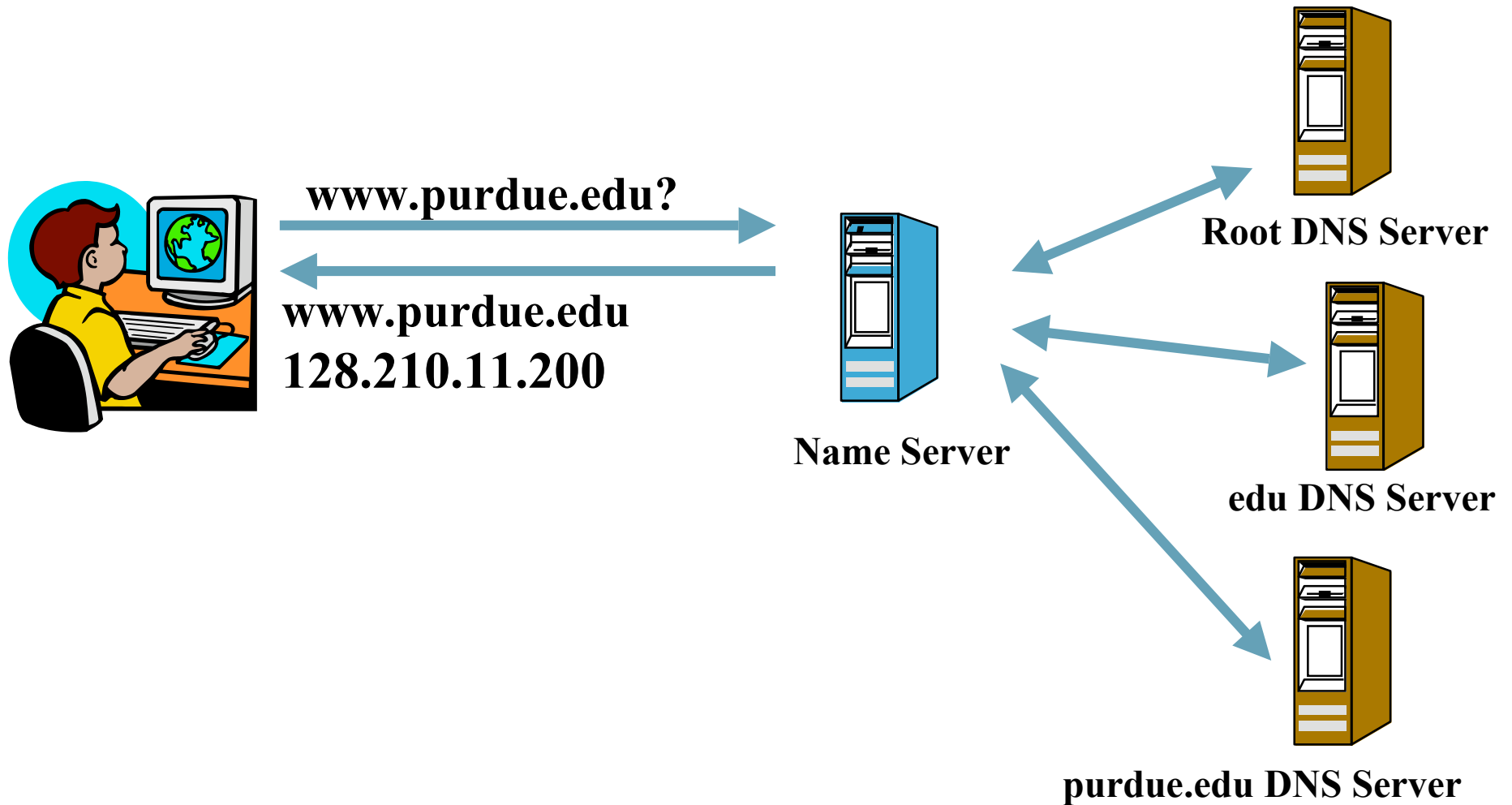
# Hierarchical Structure

- 13 root servers: **10 in USA, 1 in Sweden, 1 in UK and 1 in Japan. Why 13?**
- DNS root servers are in fact clusters of computers
- What kind of traffic? Oct 2002, 24 hours on f.root-servers.net root, 14GB, 152,744,325 queries, **1768 queries per second**
- Top Level Domain (TLD) operate “.com”, “.edu”, etc
- Name servers

# Name Server

- Each zone has a name server that maintains database of host information for its zone
- Contact the authoritative NS of that zone to get host information (such as IP)
- Information needs to be updated when host info changes in the zone
- Dynamic updates change DNS data without having to rebuild any other part of the DNS tree

# How Does it Work?



# DNS Vulnerabilities

- DNS contains no security mechanisms
- **Denial of service**: servers bombarded with requests
  - **Defective implementations** RFC1918 (private addresses) that propagate requests/updates that were not supposed to happen (blackhole servers now collect and drop this traffic)
  - **Malicious attacks**: Oct. 2002, DDoS, 9 of the root servers were affected (about 1 hour, ICMP flooding);
- Why the attack is not observable for end-users?

# How to Defend?

- Root servers are in fact clusters of machines
- Use load balancing
- Queries rate controlled, each source address is limited to a 10KBits/sec and queue size of 3 packets.
- Other ideas?

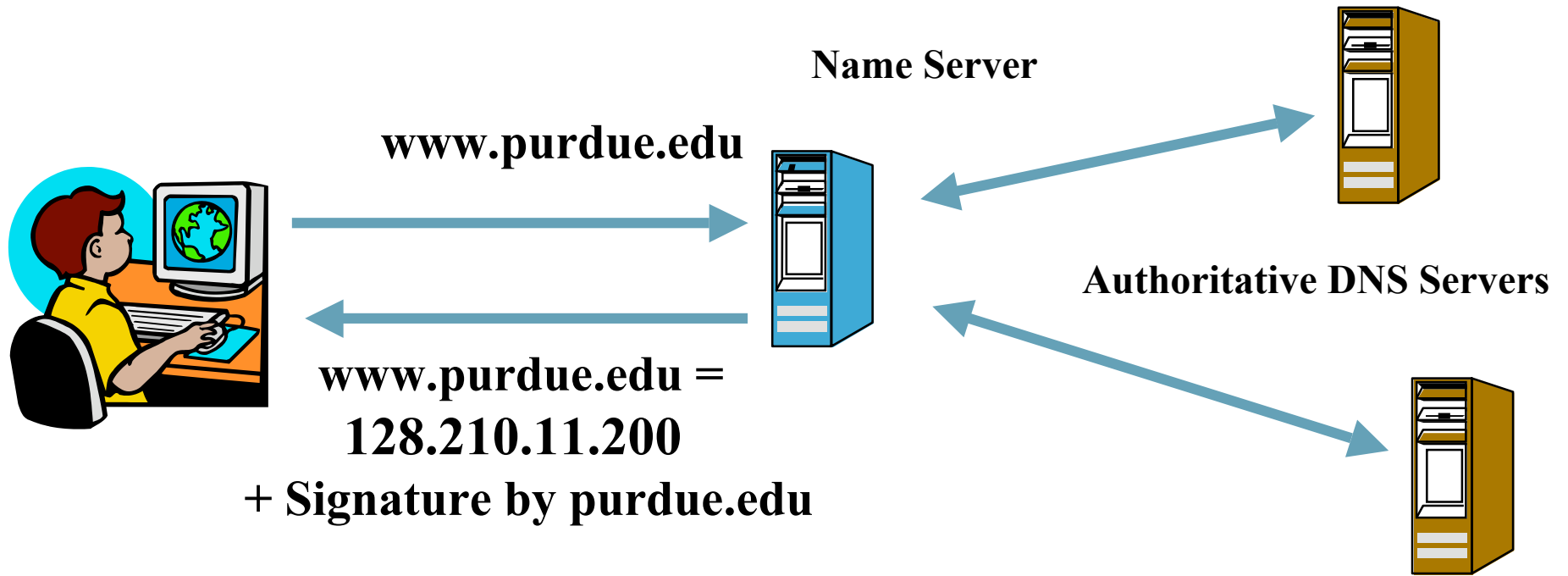
# DNS Vulnerabilities (cont.)

- **DNS Spoofing:**
  - Guessing DNS queries Ids (man in the middle)
  - Compromise the DNS servers itself
- **Cache Poisoning:** False IP with a high TTL, which the DNS server will cache for a long time
- **Email Spoofing:** Registration with ICANN often done via email and authenticated by the email address. Return addresses can be falsified
- **Mis-configuration:** Administrator enters the DNS information incorrectly

# DNSSEC

- Proposed solution: addressing authentication and integrity (digital signatures)
- Each DNS zone signs its data using its private key (signing can be done offline, in advance)
- Query for a record will return the requested resource and a digital signature of the requested resource record set
- Resolver will authenticate the response using the corresponding public key of the zone
- **WHAT THREATS DOES NOT DNSSEC address?**

# Secure DNS



# What are the Issues?

- How to obtain the public key to verify the digital signature (chicken-and-egg problem)
- Key management is critical (connected with flexibility, original design (RFC 2535) was fatally flawed because did not consider carefully key management)
- Denial of existence : prove a domain for which a query was made, does not exist
- Incremental deployment, flexible to add new domains
- Cryptography alone adds new DoS due to crypto errors and attacks

# Key Validation

- How to obtain certified public keys of zones, to verify the digital signatures
- New DNS records KEY, signed by servers in other zones
- Approaches
  - **Tree structure**: each parents signs the keys of children
  - **PGP-style web of trust**
  - **Mesh**: combination between the above, specifies how to find a path of trust

# Tree-Based Key Validation

- *A key is valid if it is signed by the parent and the parent key is valid*
- Resolvers configured to trust a master key
- The good:
  - Fits the DNS structure
  - Efficient, no problem to find path of trust
- The bad:
  - Difficult to deploy incrementally
  - Single point of failure
  - Undesirable trust relationships

# Web of Trust Key Validation

- *A key is valid if it is signed by at least one other key, any chain of trust is possible*
- The good:
  - Ease of deployment
  - No single point of failure
  - Scalable key signing
- The bad:
  - Unauthorized signatures
  - Finding a chain of trust

# Mesh-Based Key Validation

- Any zone may sign the public key of any other zone, the resolver decode which signatures are considered valid
- Each zone key has associated a routing record (lists the last link in a chain of trust);
- Default route record
- The good:
  - Ease of deployment
  - No single point of failure
  - Scalable key signing
- The bad ?

# Different Approach:SK-DNSSEC

- PKI-based solution referred as PK-DNSSEC
- For PK-DNSSEC: DNS system acts as an online Certificate Authority
- Using symmetric key cryptography would be more efficient
- Encryption and decryption are faster and require smaller keys
- Root has a globally known public key
  - All systems can authenticate communications from root
- Use symmetric key certificates build chain of trust

# Advantages

- SK signatures can be created and verified much faster than PK signatures
- PK signatures can be reused for performance, but verification is slow and must be done for every answer
- Authenticated PK-DNSSEC queries and responses don't fit into 512byte UDP datagrams, but SK-DNSSEC authoritative answers and referrals will
- SK-DNSSEC only sends 1 signature per query
- Signing a zone file in a PK-DNSSEC server increases its size by 7 times
- SK-DNSSEC gives a minimal increase (more certificates can be cached)

# Advantages

- Replay protection
  - PK-DNSSEC signatures may be replayed if the validity time is long
  - SK-DNSSEC uses nonces to prevent replay
- Possible extensions
  - Mutual authentication
  - Confidentiality
  - Can be combined with PK-DNSSEC
    - Top level domains use PK certificates
    - Lower level use SK certificates

# Summary

- DNS is a fundamental service suffering from numerous vulnerabilities
- DNSSEC proposed to provide authentication and integrity, based on digital signatures
- Main issues: deployment, how to obtain the public key to verify signatures, key rollover, authentication of denial of existence
- Not addressed
  - Denial of service
  - Mis-configuration, validation of the content