CIS 433/533: Computer and Network Security
DNS and IP Security

Prof. Kevin Butler
Winter 2013
Today’s Class

- Look-ahead to exam
- Discuss attacks against DNS
- Understand DNSSEC
- Look at IPsec and its operation
- VPN and VLAN
Exam Prep

- I sent out a list of chapters from the book that will have covered by today’s class
- Review questions from each chapter are good
- List of good problems to improve skills has been sent out over email
- Exam will cover:
  - material from lectures (slides and board notes/discussion)
  - material from textbook (relevant chapters noted)
  - material from papers read for class
Exam format

• 80 minutes, 100 points available
• Short answer questions (40 points): primarily knowledge of terms and concepts
• Long answer questions (30 points): more deeply explain particular concepts, perhaps with examples or more formal notation
• Word and construction problems (30 points): detailed understanding of existing ideas, application of skills to new types of problems
Allowed/Expected Materials

• You are allowed one 8 1/2 x 11 crib sheet, double-sided
• You can bring a non-programmable calculator
  ‣ smartphones acceptable only if calculator application alone is displayed on your phone: turn to airplane mode
• Bring your student ID and have it out on your desk
• Stagger your seating so that you are not directly sitting next to anyone else
• Exam ends exactly 80 minutes after it begins
Project Related Work

• Reminder: related work is due today for project

• Grading will consider the citations you have found and the quality of the section

• Next class after midterm: will discuss more about structuring the project and paper/poster
• DNS maps between IP address (128.223.4.25) and domain and host names (www.cs.uoregon.edu)
  ‣ How it works: the “root” servers redirect you to the top level domains (TLD) DNS servers, which redirect you to the appropriate sub-domain, and recursively ....
  ‣ Note: there are 13 “root” servers that contain the TLDs for .org, .edu, and country specific registries (.fr, .ch)
A DNS query

(1) www.kevinbutler.org?

DNS Cache
www.kevinbutler.org = 184.168.221.24

ISP Nameserver

User PC

(2) www.kevinbutler.org?
(3) redirect

(4) www.kevinbutler.org?
(5) redirect

(6) www.kevinbutler.org?

(7) 184.168.221.24

(8) 184.168.221.24
“Glue” information

• Suppose you ask a name server for a record and it redirects you to another name server (NS record)
  ‣ e.g., if you ask a root for a NS (name server) record for NET, it returns NS records for the authoritative servers for .net

• It will also give you the A (resource) record for the authoritative servers you were directed to
  ‣ avoid looking them up
  ‣ These are known as “glue” records
DNS Vulnerabilities

• Nothing is authenticated, so really the game is over
  ‣ You can not really trust what you hear …
  ‣ But, many applications are doing just that.
  ‣ Spoofing of DNS is really dangerous

• Moreover, DNS is a catalog of resources
  ‣ Zone-transfers allow bulk acquisition of DNS data
  ‣ … and hence provide a map for attacking the network

• Lots of opportunity to abuse the system
  ‣ Relies heavily on caching for efficiency -- cache poisoning
  ‣ Once something is wrong, it can remain that way in caches for a long time (e.g., it takes a long time flush)
  ‣ Data may be corrupted before it gets to authoritative server
Ghost Domain Names

• Jun Li et al., NDSS 2011
• Malicious domain names can be resolved long after they have been removed from TLD and other upper-level DNS servers
• Attacker can renew cached delegation data even after data has been deleted from parent (extend TTL value in the record)
• Lots of vulnerable DNS implementations and public servers
A Cache Poisoning Attack

• All requests have a unique query ID

• The nameserver/resolver uses this information to match up requests and responses

• If an adversary can guess the query ID, then it can forge the responses and pollute the DNS cache
  ‣ 16-bit query IDs (not hard)
  ‣ Some servers increment IDs (or use other bad algo.)
  ‣ First one in wins!!!

• Note: If you can observe the traffic going to a name server, you can pretty much arbitrarily own the Internet for the clients it serves.
1. Query a random host in a victim zone, e.g., 1234.cs.uoregon.edu

2. Spoof responses* as before, but delegate authority to some server which you own.
   1. The glue records you give make you authoritative

3. You now own the domain.

*the original attack exploited poor ID selection
Kaminsky Fixes

- Make the ID harder to guess (randomized ports)
  - Amplified ID space from $2^{16}$ to $2^{27}$
- Prevent foreign requests from being processed
  - E.g., filter requests from outside domain
- Observe and filter conflicting requests
  - E.g., if you see a lot of bogus looking requests, be careful
- All of this treats the symptoms, not the disease.
  - Lack of authenticated values
  - Thus, if you can observe request traffic, prevent legitimate responses, or are just plain patient, you can mount these attacks.
A standard-based (IETF) solution to security in DNS

- Prevents data spoofing and corruption
- Public key based solution to verifying DNS data
- Authenticates
  - Communication between servers
  - DNS data
    - content
    - existence
    - non-existence
  - Public keys (a bootstrap for PKI?)
DNSSEC Mechanisms

• TSIG: transaction signatures protect DNS operations
  ‣ Zone loads, some server to server requests (master -> slave), etc.
  ‣ Time-stamped signed responses for dynamic requests
  ‣ A misnomer -- it currently uses shared secrets for TSIG (HMAC) or do real signatures using public key cryptography

• SIG0: a public key equivalent of TSIG
  ‣ Works similarly, but with public keys
  ‣ Not as popular as TSIG, being evaluated

• Note: these mechanisms assume clock sync. (NTP)
• Securing the DNS records
  ‣ Each domain signs their “zone” with a private key
  ‣ Public keys published via DNS
  ‣ *Indirectly* signed by parent zones
  ‣ Ideally, you only need a self-signed root, and follow keys down the hierarchy
DNSSEC challenges

• Incremental deployability
  ‣ Everyone has DNS, can’t assume a flag day

• Resource imbalances
  ‣ Some devices can’t afford real authentication

• Cultural
  ‣ Most people don’t have any strong reason to have secure DNS ($$$ not justified in most environments)
  ‣ Lots of transitive trust assumptions (you have no idea how the middlemen do business)

• Take away: DNSSEC will be deployed, but it is unclear whether it will be used appropriately/widely
Transport Security

• A host wants to establish a secure channel to remote hosts over an untrusted network
  ‣ Not Login – end-users may not even be aware that protections in place (transparent)
  ‣ Remote hosts may be internal or external

• The protection service must …
  ‣ Authenticate the end-points (each other)
  ‣ Negotiate what security is necessary (and how achieved)
  ‣ Establish a secure channel (e.g., key distribution/agreement)
  ‣ Process the traffic between the end points

• Also known as communications security.
IPsec

• Host level protection service
  ‣ IP-layer security (below TCP/UDP)
  ‣ De facto standard for host level security
  ‣ Developed by the IETF (over many years)
  ‣ Available in most operating systems/devices
    • E.g., XP, Vista, OS X, Linux, BSD*, …
  ‣ Implements a wide range of protocols and cryptographic algorithms

• Selectively provides …. 
  ‣ Confidentiality, integrity, authenticity, replay protection, DOS protection
IPsec and the IP protocol stack

• IPsec puts the two main protocols in between IP and the other protocols
  ‣ AH - authentication header
  ‣ ESP - encapsulating security payload

• Other functions provided by external protocols and architectures
Modes of operation

- **Transport**: the payload is encrypted and the non-mutable fields are integrity verified (via MAC)

- **Tunnel**: each packet is completely encapsulated (encrypted) in an outer IP packet
  - Hides not only data, but some routing information
Tunneling

• “IP over IP”
  ‣ Network-level packets are encapsulated
  ‣ Allows traffic to avoid firewalls
IPsec Protocol Suite

Policy/Configuration Management

(SPS) Security Policy System

Key Management

(IKE) Internet Key Exchange

Manual

Packet Processing

(ESP) Encapsulating Security Payload

(AH) Authentication Header
Internet Key Exchange (IKE)

• Two phase protocol used to establish parameters and keys for session
  ‣ Phase 1: authenticate peers, establish secure channel
  ‣ Phase 2: negotiate parameters, establish a security association (SA)

• The details are unimaginably complex

• The SA defines algorithms, keys, and policy used to secure the session

• Simplified successors: IKEv2, Just Fast Keying (JFK)
IPsec: Packet Handling (Bump ...)

IP Protocol Stack

- Application
- Presentation
- Session
- Transport
- Network (IP)
- Data Link
- Physical

SADB

IPsec
Authentication Header (AH)

• Authenticity and integrity
  ‣ via HMAC
  ‣ over IP headers and data

• Advantage: the authenticity of data and IP header information is protected
  ‣ it gets a little complicated with *mutable* fields, which are supposed to be altered by network as packet traverses the network
  ‣ some fields are *immutable*, and are protected

• Confidentiality of data is *not* preserved

• Replay protection via AH sequence numbers
  ‣ note that this replicates some features of TCP (good?)
Authentication Header (AH)

- Modifications to the packet format
IPsec AH Packet Format

IPv4 AH Packet Format

| IPv4 Header | Authentication Header | Higher Level Protocol Data |

AH Header Format

<table>
<thead>
<tr>
<th>Next Header</th>
<th>Length</th>
<th>Reserved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Parameter Index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication Data (variable number of 32-bit words)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IPsec Authentication

- **SPI**: (spy) identifies the security association for this packet
  - Type of crypto checksum, how large it is, and how it is computed
  - Really the policy for the packet

- **Authentication data**
  - Hash of packet contents include IP header as specified by SPI
  - Treat transient fields (TTL, header checksum) as zero

- **Keyed MD5 Hash is default**
Encapsulating Security Payload (ESP)

- Confidentiality, authenticity and integrity
  - via encryption and HMAC
  - over IP \texttt{payload} (data)
- Advantage: the security manipulations are done solely on user data
  - TCP packet is fully secured
  - simplifies processing
- Use “null” encryption to get authenticity/integrity only
- Note that the TCP ports are hidden when encrypted
  - good: better security, less is known about traffic
  - bad: impossible for FW to filter/traffic based on port
- Cost: can require many more resources than AH
Encapsulating Security Payload (ESP)

- Modifications to packet format
**IPsec ESP Packet Format**

**IPv4 ESP Packet Format**

<table>
<thead>
<tr>
<th>Unencrypted</th>
<th>Encrypted</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Header</td>
<td>Other IP Headers</td>
</tr>
</tbody>
</table>

**ESP Header Format**

- Security Parameter Identifier (SPI)
- Opaque Transform Data, variable length

**DES + MD5 ESP Format**

- Security Parameters Index (SPI)
- Initialization Vector (optional)
- Replay Prevention Field (incrementing count)
- Payload Data (with padding)
- Authentication checksum
Practical Issues and Limitations

• IPsec implementations
  ‣ Large footprint
    • resource poor devices are in trouble
    • New standards to simplify (e.g., JFK, IKE2)
  ‣ Slow to adopt new technologies
  ‣ Configuration is really complicated/obscure

• Issues
  ‣ IPsec tries to be “everything for everybody at all times”
    • Massive, complicated, and unwieldy
  ‣ Policy infrastructure has not emerged
  ‣ Large-scale management tools are limited (e.g., CISCO)
  ‣ Often not used securely (common pre-shared keys)
Network Isolation: VPNs

• Idea: I want to create a collection of hosts that operate in a coordinated way
  ‣ E.g., a virtual security perimeter over physical network
  ‣ Hosts work as if they are isolated from malicious hosts

• Solution: Virtual Private Networks
  ‣ Create virtual network topology over physical network
  ‣ Use communications security protocol suites to secure virtual links “tunneling”
  ‣ Manage networks as if they are physically separate
  ‣ Hosts can route traffic to regular networks (split-tunneling)
VPN Example: RW/Telecommuter

Internet

LAN

Physical Link
Logical Link (IPsec)

(network edge)
VPN Example: Hub and Spoke

Internet

LAN

(Network edge)

Physical Link

Logical Link (IPsec)
VPN Example: Mesh

Internet

LAN

(network edge)

Physical Link

Logical Link (IPsec)
Virtual LANs (VLANs)

- VPNs built with hardware
  - Physically wire VPN via soft configuration of a switch crossbar
  - No encryption – none needed
  - “wire based isolation”
  - Many switches support VLANs
  - Allows networks to be reorganized without rewiring
- Example usage: two departments in same hallway
  - Each office is associated with department
  - Configuring the network switch gives physical isolation
  - Note: often used to ensure QoS