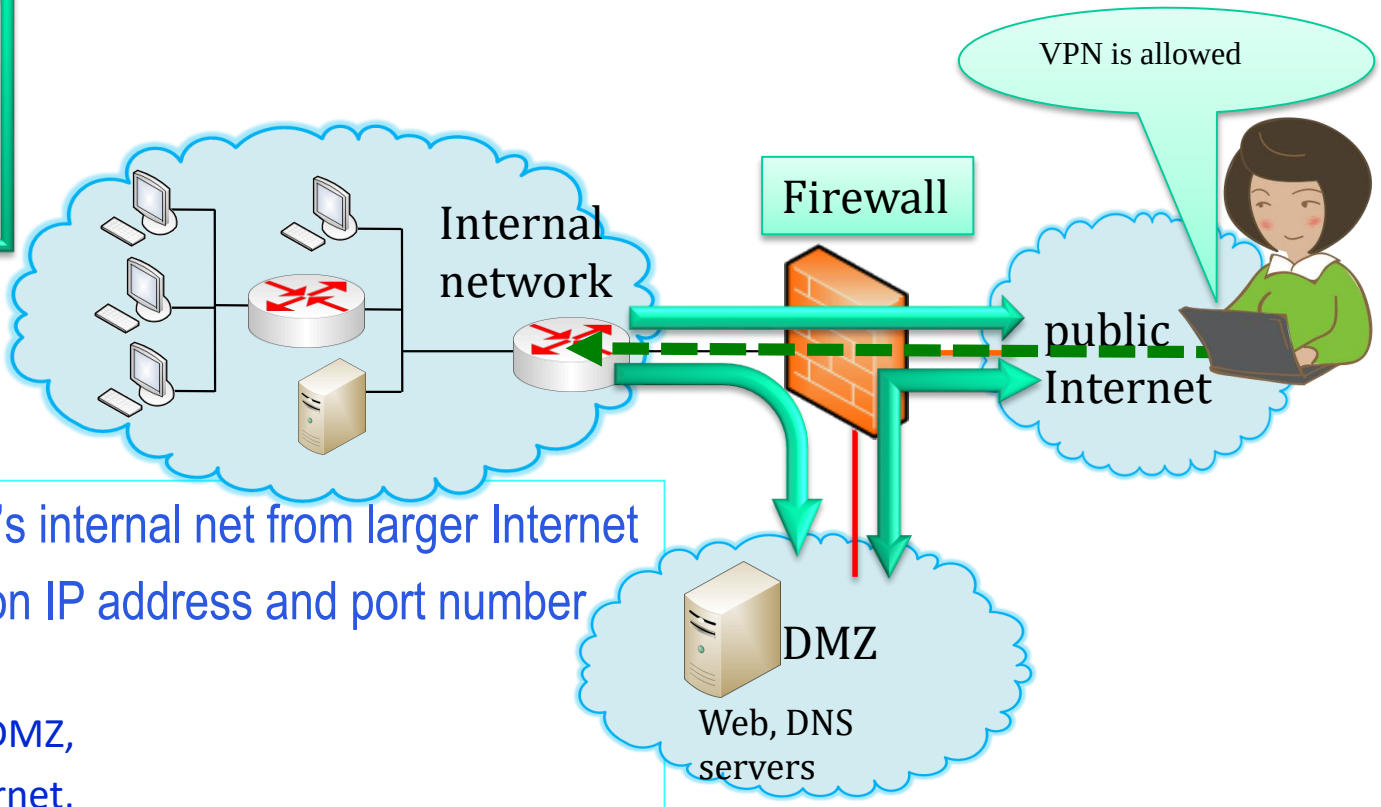
Lecture 6

Network Security

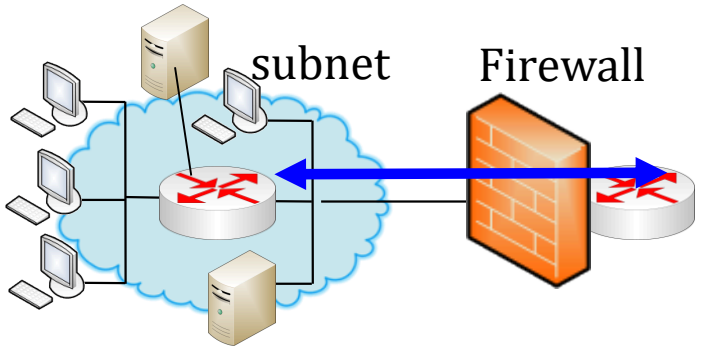
Firewalls



- Isolate organization's internal net from larger Internet
- Allow/Block based on IP address and port number
  - Allow:
    - Internet to DMZ,
    - DMZ to Internet,
    - Internal to Internet
  - Block:
    - Internet to Internal, DMZ to Internal
    - DMZ: demilitarized zone

Firewall Locations in the Network

- Between internal network and external network
- At the gateways of sensitive subnetworks within the internal network
  - Payroll's network must be protected separately within the corporate network
  - Dual-homed firewall
- On end-user machines
  - Personal firewall
  - Microsoft's Internet Connection
  - Firewall (ICF) comes standard
  - In Windows XP and Windows 7/Vista



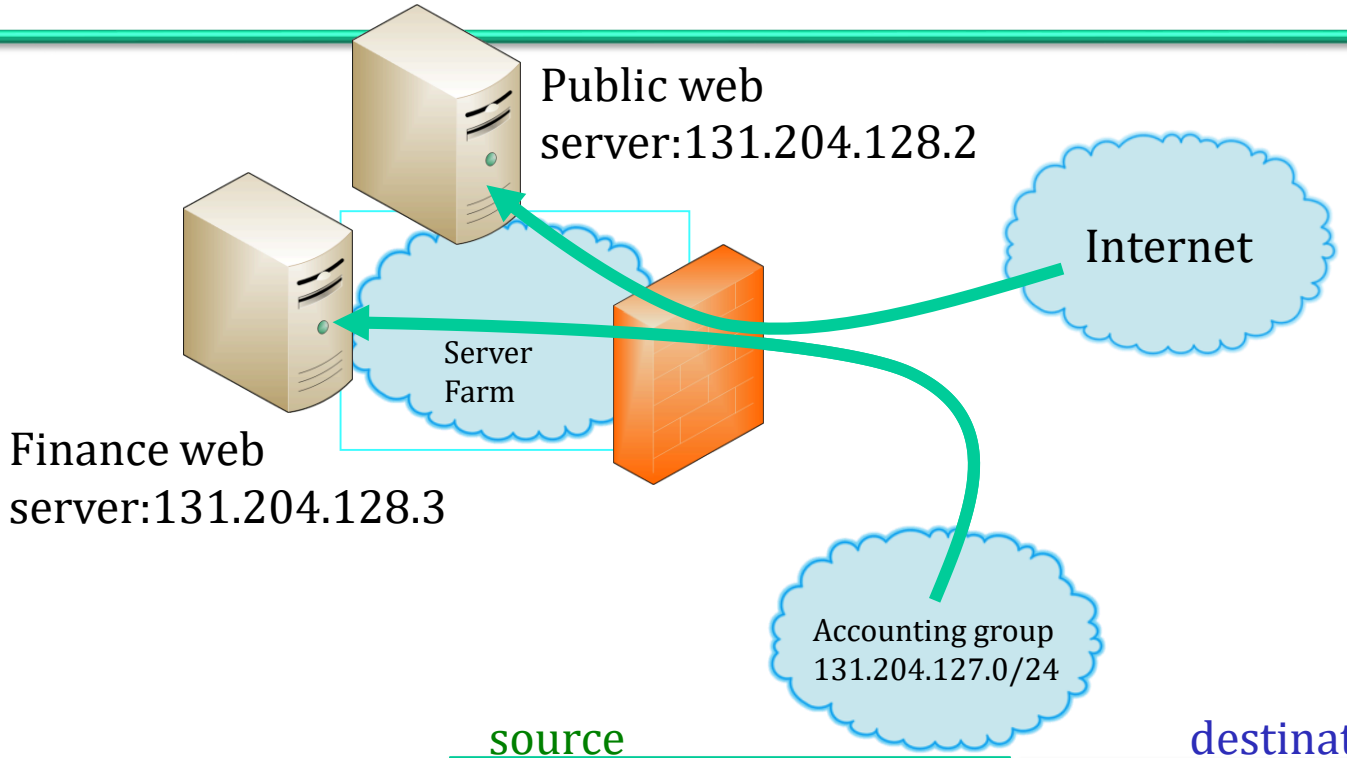
Firewall Types

- Packet filtering
  - Stateful inspection (Session filter)
  - Stateless inspection
- Proxy gateway
  - All incoming traffic is directed to firewall and then passed to client after inspection
  - All outgoing traffic is directed to firewall and then sent by firewall after inspection
  - Application-level: inspect content (payload) of packets
    - Separate proxy for SMTP (email), HTTP, FTP, etc.
    - Filtering rules are application-specific
  - Circuit-level: application-independent, “transparent” to user
    - Only generic IP traffic filtering (example: SOCKS)
    - No contect inspection
- Personal firewall with application-specific rules
  - Allows specific applications to pass through firewall
  - E.g., https (port 443), DNS (port 53)

Stateless Packet Filtering

- For each packet, firewall decides whether to allow it to proceed
  - Decision must be made on per-packet basis
    - Stateless
    - No inspection for packet context
      - TCP connection/application
- Use packet header for inspection
  - IP source and destination addresses, ports
  - Protocol identifier (TCP, UDP, ICMP, etc.)
  - TCP flags (SYN, ACK, RST, PSH, FIN)
  - ICMP message type
- Filtering rules are based on pattern-matching
- Filtering rules are also called the Access control list (ACL) in Cisco and other vendors

Packet filtering example



https

source

destination

access-list 100 permit tcp host 131.204.127.0/24 gt 1023 131.204.128.3 eq 443
access-list 100 permit tcp any gt 1023 host 131.204.128.2 eq 80

interface Ethernet 0/0
ip access-group 100 in

Anything not explicitly permitted by the access list is denied!

Inbound packet and outbound packet filtering

- Out: Traffic that has already been through the router and leaves the interface. The source is where it has been, on the other side of the router, and the destination is where it is going.
- In: Traffic that arrives on the interface and then goes through the router. The source is where it has been and the destination is where it is going, on the other side of the router.
- Inbound :
  - If the access list is inbound, when the router receives a packet, the Cisco IOS software checks the criteria statements of the access list for a match
  - If the packet is permitted, the software continues to process the packet. If the packet is denied, the software discards the packet.
- Outbound:
  - If the access list is outbound, after the software receives and routes a packet to the outbound interface, the software checks the criteria statements of the access list for a match
  - If the packet is permitted, the software transmits the packet. If the packet is denied, the software discards the packet.
-

Access control list (ACL)

- An ACL consists of one or more entries
  - The ACL is a sequential collection of permit and deny conditions
  - Depending on the ACL type, one can specify a match criteria using the source and destination addresses, the protocol, the ports (for TCP or UDP), the ICMP type, ICMP code, or the EtherType
- The order of the entries is important
  - When a firewall decides whether to accept or refuse a connection, the firewall tests the packet against each ACL in the order in which the entries are listed
  - After it finds a match, the firewall does not check any more entries
    - For example, if one creates an entry at the beginning of an ACL that explicitly permits all traffic, the firewall does not check any further statements in the ACL
  - All ACLs have an implicit deny entry at the end of the ACL
  - Unless one explicitly permits it, traffic cannot pass through the firewall
    - For example, if one wants to allow all users to access a network through the firewall except for those with particular IP addresses, then one needs to deny the particular IP addresses in one entry and then permit all other IP addresses in another entry

Weaknesses of Packet Filters

- Do not prevent application-specific attacks
  - No content (payload) inspection
  - For example, firewall will not block an attack string that is a buffer overflow in a URL decoding routine
- No user authentication mechanisms
  - Only (spoofable) address-based authentication
  - Solution: list of addresses for each router interface
    - Packets with internal addresses should not come from outside

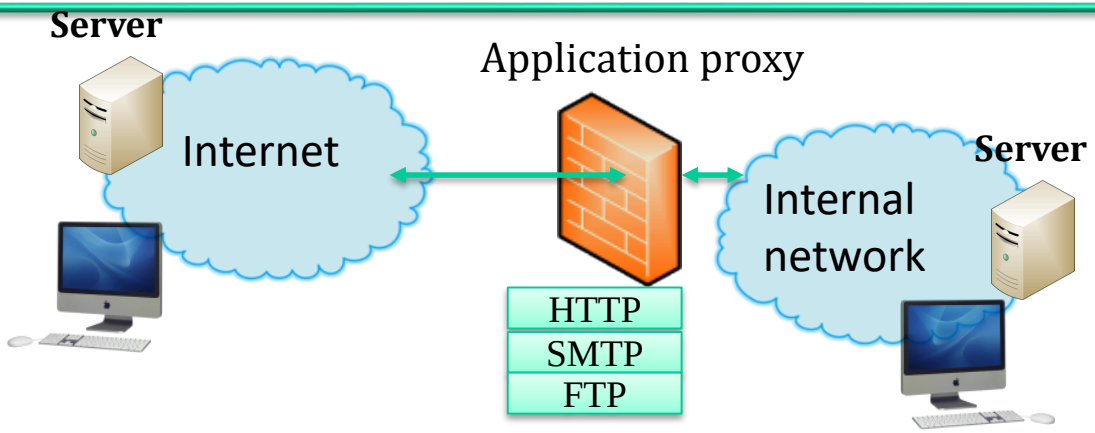
Stateless Port Filtering Is Useless

- In TCP connections, ports with numbers less than 1024 are permanently assigned to servers
- Clients use ports numbered from 1024 to 16383
- When a firewall processes an incoming packet to some client's port 5612, it will not block this packet since
  - This packet could be a server's response in a previously established connection
  - OR it could be malicious traffic
  - It is necessary to keep state for each connection
- Stateful inspection (session) packet filtering firewall

Stateful/Session Filtering

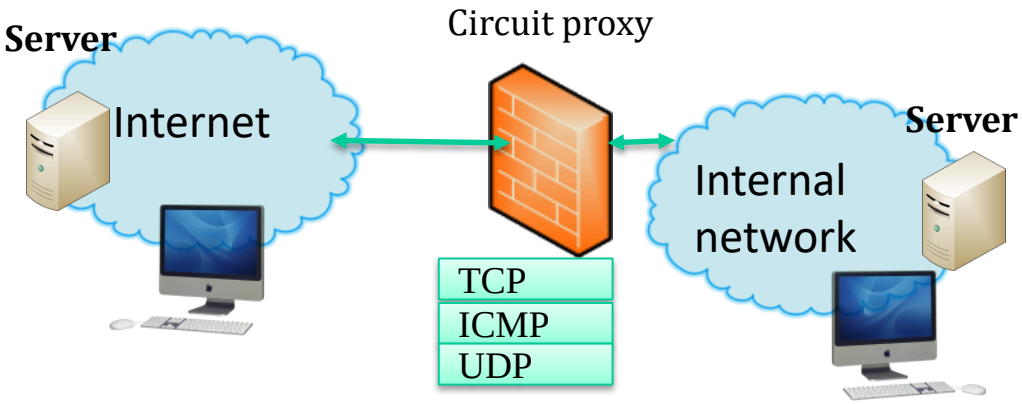
- Filtering for each packet is based on the context of a connection
  - If new connection, then check against security policy
  - If existing connection, then look it up in the state transition table and update the table, if necessary
    - Only allow incoming traffic to a high-numbered port if there is an established connection to that port
- Challenges: stateless protocols
  - UDP and ICMP
- Default filter: deny everything that is not explicitly permitted
  - Block ICMP unless debugging is needed
- The filtering rules (ACL) has the same format as stateless filtering
- Filters can be bypassed with VPN using IP tunneling

Application-Level Gateway



- Inspect and relay application-specific connections
  - http/SMTP/FTP proxy
  - Big overhead and slow
  - Can filter content, log and audit all activity
- Support user-to-gateway authentication
- Need separate proxy for each application
- Example: Microsoft ISA, SQUID

Circuit-Level Gateway



- Inspects and relays TCP/UDP/ICMP
  - Based on the state (session) for filtering
  - Authentication provides the basis for filtering
  - Does not inspect the contents of segments
  - Weaker than application-level gateway
  - Faster than application-level gateway
- Combined proxy: for lower overhead
  - Application-level proxy on inbound for protecting critical servers
  - Circuit-level on outbound (trusted users)

SOCKS (1)

- SOCKS is an abbreviation for "SOCKetS"
- Circuit level Gateway
- SOCKS performs at Layer 5 of the OSI model
  - The Session Layer above transport layer
- SOCKS v5 (RFC 1928)
  - Support TCP
  - Supports UDP and ICMP, earlier versions did not
  - Provides strong user authentication, and host name resolution (RFC 1929)
- SOCKS clients: Proxifier, WideCap, ProxyCap, FreeCap, Hummingbird SOCKS

SOCKS (2)

- Clients, behind a firewall, need to access exterior servers
  - Connect to a SOCKS proxy server
  - Perform authentication
- Proxy server
  - Controls the eligibility of the client to access the external server using access control list
  - Passes the request on to the server if the request is allowed
- SOCKS can also be used in the opposite direction
  - Allowing the clients outside the firewall ("exterior clients") to connect to servers inside the firewall (internal servers)
  - Based on access control list

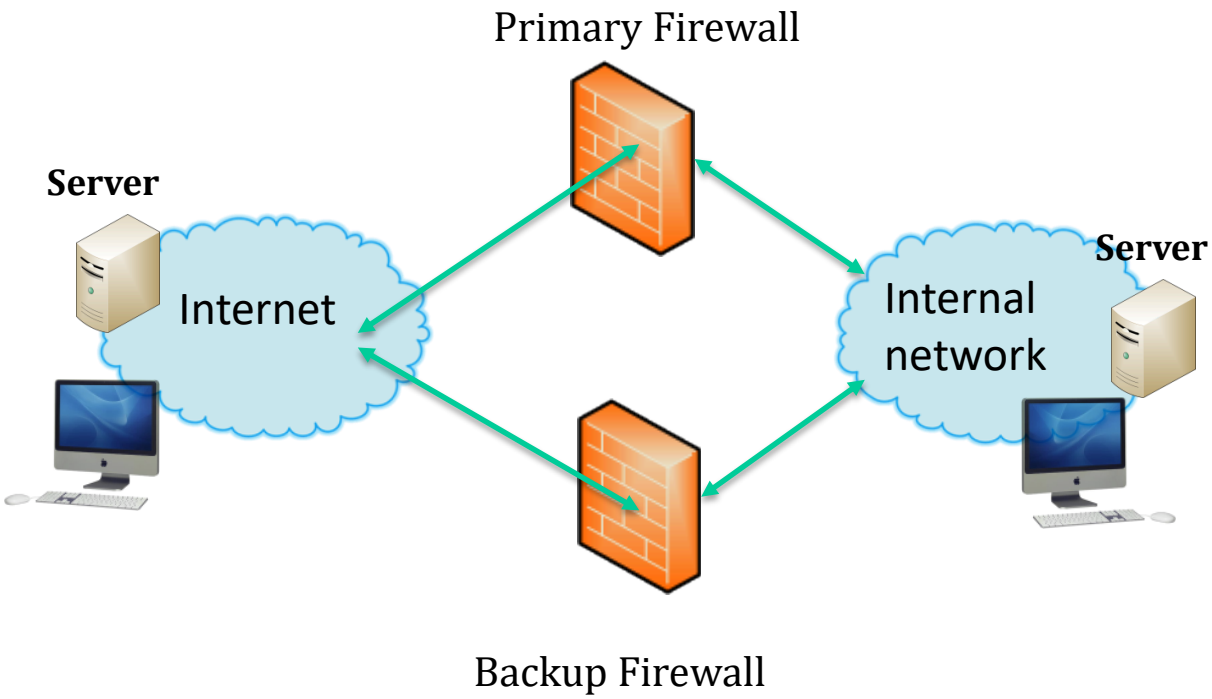
Protecting Addresses, Services and Routes

- Hide IP addresses and services of the hosts on an internal network
  - Only services that are intended to be accessed from outside need to reveal their IP addresses in the firewall
  - Keep other addresses secret to make spoofing harder
  - Use NAT (network address translation) to map addresses in packet headers to internal addresses
    - 1-to-1 or N-to-1 mapping
- Filter route announcements
  - No need to advertise routes to internal hosts
  - Prevent attacker from advertising that the shortest route to an internal host lies through him
- Disable ICMP in the firewall

Firewall weakness

- No content inspection causes the problems
  - Software weakness (e.g. buffer overflow, and SQL injection exploits)
  - Protocol weakness (WEP in 802.11)
- No defense against
  - Denial of service
  - Insider attacks

Primary-backup firewall architecture

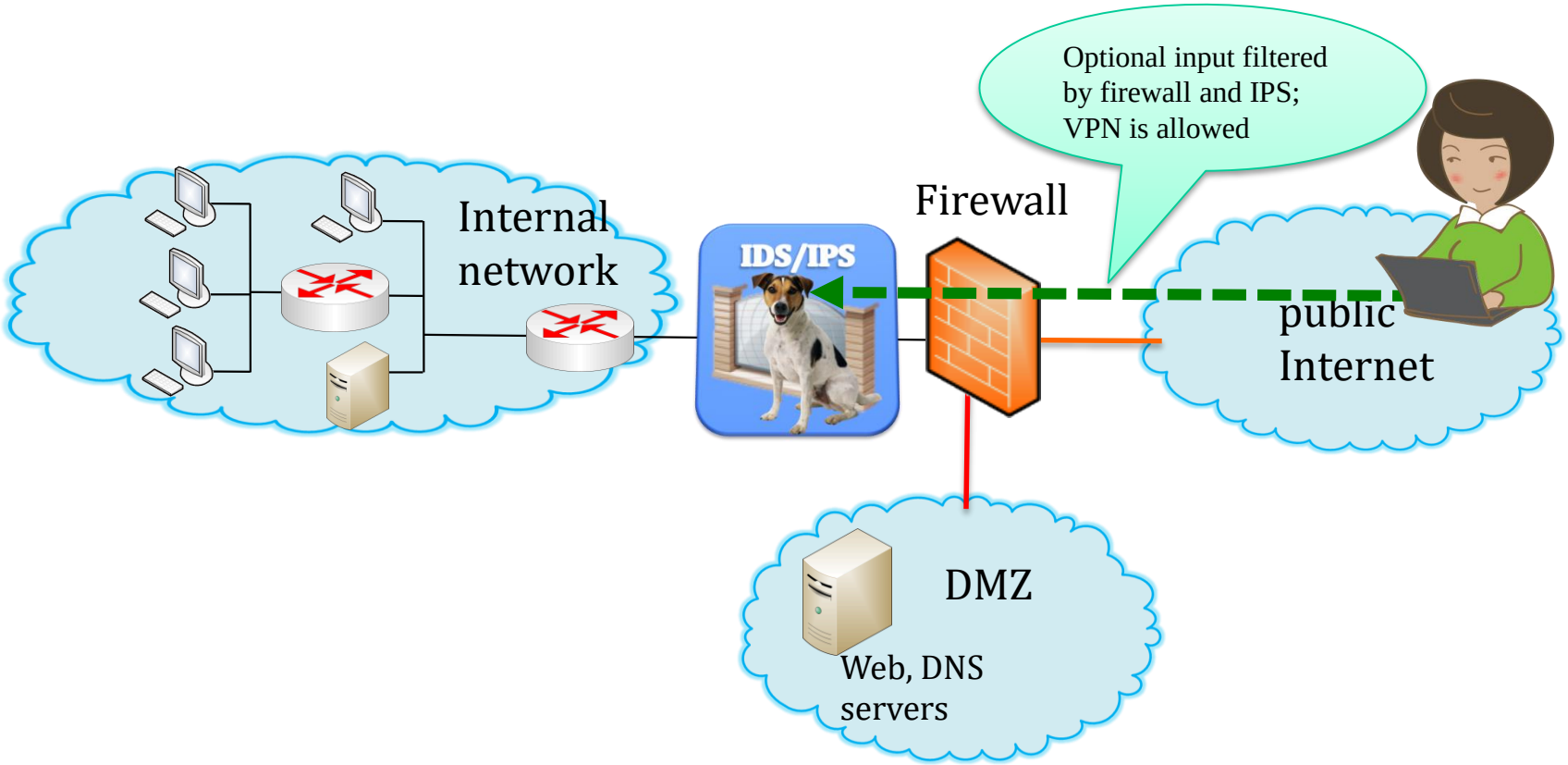


Windows 7/Vista Firewall

- Supports filtering for both incoming and outgoing traffic
- Microsoft allows rule creation based on
  - Application,
  - TCP and UDP port
  - Interface
  - IP Address
  - ICMP protocol, allowing filtering by ICMP type
- Three separate groups: input, output and connection security
  - The input and output groups are used for input and output packets, respectively.
  - The connection security group allows authentication between two devices and communications via Ipsec
- Default behavior of Windows 7/Vista Firewall:
  - Block all incoming traffic unless it is solicited or it matches a configured rule
  - Allow all outgoing traffic unless it matches a configured rule

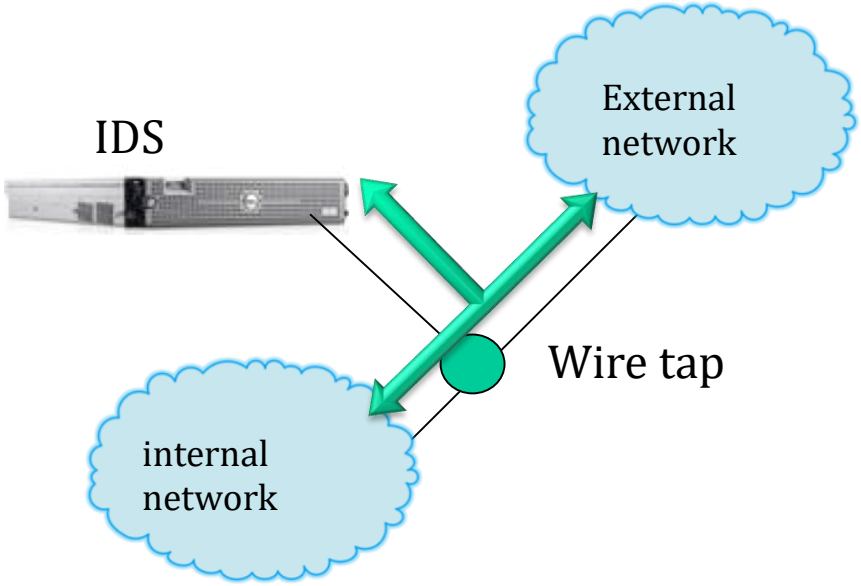
Intrusion Detection/Prevention system (IDS/IPS)

- Deep packet inspection (payload)
- IDS: report intrusions by out of band detection
- IPS: Block intrusions by in band filtering

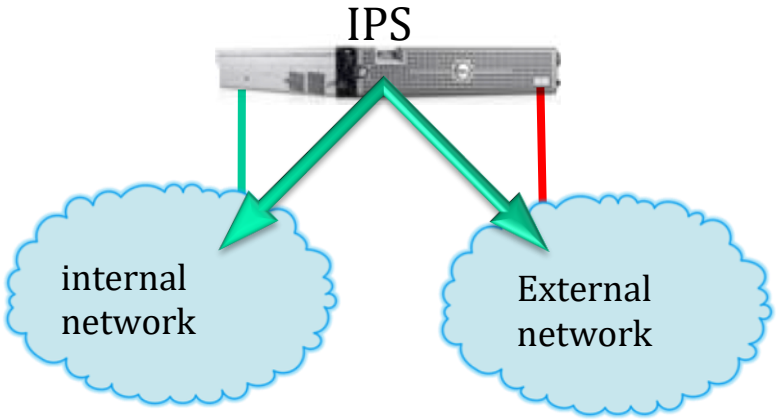


IDS vs. IPS

IDS: out of band



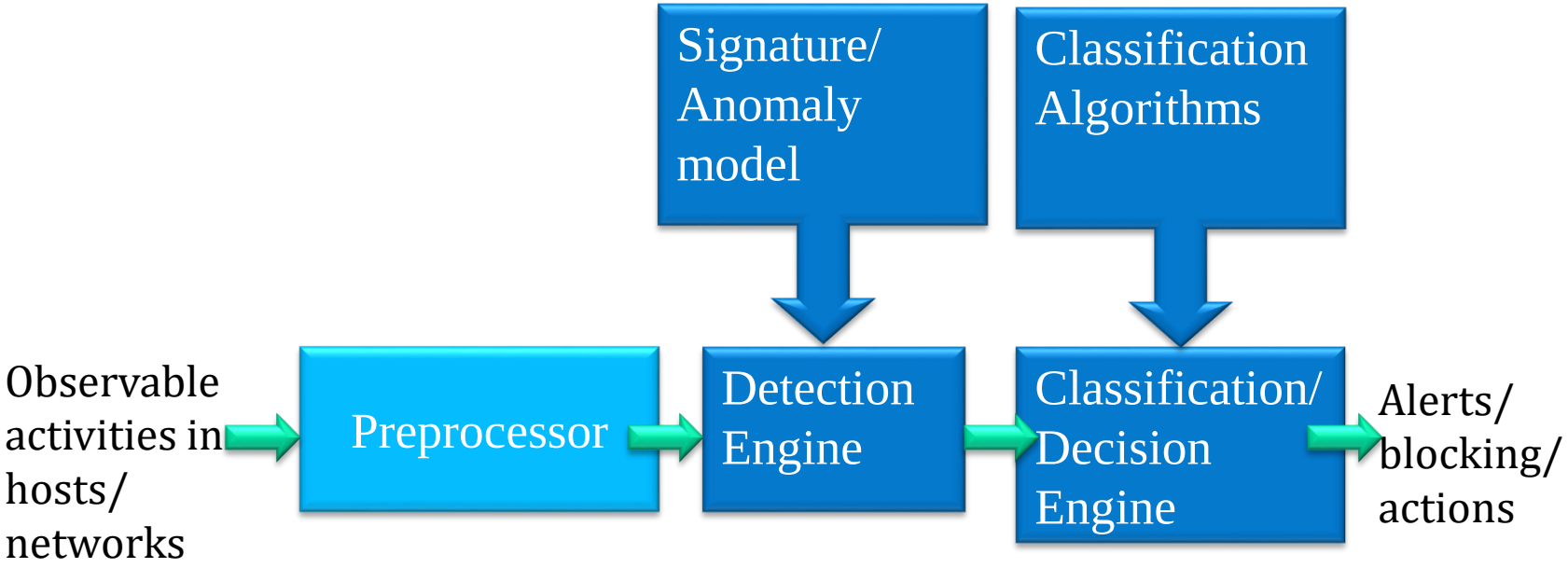
IPS: in line



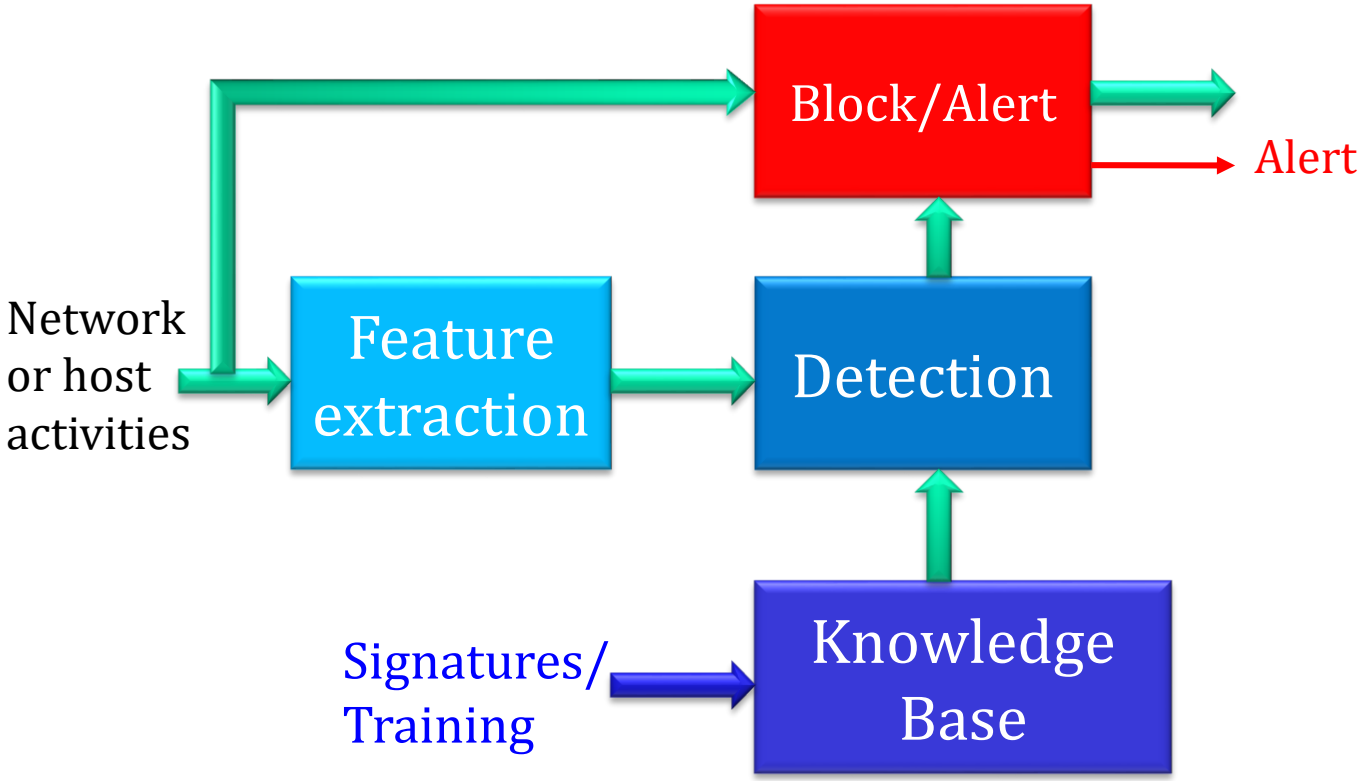
Intruders

- Aim to gain access of resources and/or increase privileges on an information infrastructure
  - Flaws in system/networks that could be exploited to violate the security policy of a system or network
- Awareness of intruders has led to the development of CERTs
  - The United States Computer Emergency Readiness Team (US-CERT.gov) alerts
- Use compromised system to launch other attacks
- Basic attack methodology
  - Target acquisition and information gathering (Reconnaissance and Scanning)
  - Gaining Access
  - Privilege escalation
  - Maintaining Access
  - Covering tracks

IDS/IPS block diagram



IDS/IPS Function Blocks



IDS/IPS

- Host-based IDS/IPS
  - Monitor/block activity on a single host
  - Advantage: better visibility into behavior of individual applications running on the host
- Network-based IDS/IPS (NIDS/NIPS)
  - Often placed behind a router or firewall that is the entrance of a critical asset
  - Monitor traffic, and examine packet headers and payloads
  - Advantage: single NIDS/IPS can protect many hosts and detect global patterns
- IDS needs to detect a substantial percentage of intrusions with few false alarms
  - If too few intrusions detected (false negatives): no security
  - If too many false alarms (false positives): ignore
  - Existing systems are improving

Intrusion Detection Approaches

- Behavior-based detection
  - Statistics-based anomaly detection
    - Threshold independent of user
    - Profile per user
    - Computationally expensive
  - Rule-based suspicious behavior detection
    - Behavior is commonly associated with a type of attack, such as buffer overflow
  - Pro: May detect zero-day attacks
  - Con: Usually higher false positive rate
- Signature-based detection
  - Detect patterns of specific known exploits and vulnerabilities
    - Exploits: patterns of codes, scripts, registration key modification, RET addresses of buffer overflow exploits
    - Vulnerabilities: traffic or requests to a known vulnerability
      - Vulnerability-based signatures
      - Once a new vulnerability is disclosed, researchers develop signatures that anticipate the nature of yet-to-be-created threats
  - Signature captures
  - Pro: Usually lower false positive rate
  - Con: May not detect zero-day attacks

DoS/DDoS detection

- Threshold-based detection,
  - Network security managers can utilize pre-programmed limits on data traffic to ensure servers will not become unavailable due to overload
- Self-learning methodologies
  - Learn the patterns of network usage and traffic
  - Understand the wide variety of lawful, though unusual, usage patterns that may take place during legitimate network operations
- The combination of the two yields the highest accuracy of detection for DoS/DDoS attacks

IDS vs. IPS (1)

- IDS: out of band
  - An IDS false positive is an alert that did not result in an intrusion
    - The system under attack was not vulnerable to the attack
    - The detection mechanism may be faulty
    - IDS detected an anomaly that turned out to be benign
  - An IDS false positive causes a security analyst to expend unnecessary effort
    - Minimize false positives
  - No interference with traffic

- IPS: in band
  - When an IPS has a false positive, legitimate traffic will be blocked
  - IPS cannot have false positives
  - Better development for filters and thorough tests
  - To match the line speed, IPS hardware requirement is higher
    - ASIC or FPGA

IDS vs. IPS (2)

- IDS filters create leads on suspicious activity intended for a expert to follow
- IPS filters are used for automatic action such as blocking traffic or quarantining an endpoint
- Anomaly-based detection mechanisms (both protocol and statistical) are useful for IDS, but inappropriate for IPS
- Anomaly filters cannot be used for blocking, only for alerting

- Legitimate traffic in real networks contains anomalies
  - Protocol anomalies come from custom applications that use off-the-shelf protocol libraries, but use them in unexpected ways
- Behavioral anomalies come from exceptional, but often critical, business processes

Honeypots

- Filled with fabricated/counter-intelligence information
- Decoy systems to lure attackers or counter spammers
  - A way from accessing critical systems
  - Collect forsensic information on attackers activities
  - Signature extraction for IDS/IPS
  - To encourage an attacker to stay on system so an administrator can respond
- Single or multiple networked systems
- Available open source honeypots:
  - Honeyd: http://www.honeyd.org/
    - Many different hosts running different services can be simulated
    - Up to 65536 hosts
    - By emulating computers on the unused IP address of a network
  - Mwcollect: http://www.mwcollect.org/
    - Great for collecting signatures
    - Daemons obtain the malware binaries from the exploit payload using known patterns
    - The whole exploitation process is simulated in a virtualized environment to avoid being infected by malware

Open source NIDS/NIPS

- Snort is a free and open source
  - Snort was written by Martin Roesch
  - Now developed by Sourcefire, of which Roesch is the founder and CTO
  - The most widely deployed intrusion detection and prevention technology worldwide
- Using a rule-driven language
  - Combining the benefits of signature, protocol and anomaly based inspection methods.
  - Snort can be combined with other software such as SnortSnarf, sguil, OSSIM, and the Basic Analysis and Security Engine (BASE) to provide a visual console
  - Emerging Threats: Community maintained Snort rule sets are evolving
- Large rule sets for known vulnerabilities

Snort rule example (1)

Rule header contains the rule's action, protocol, source and destination IP addresses and netmasks, and the source and destination ports information

Rule header

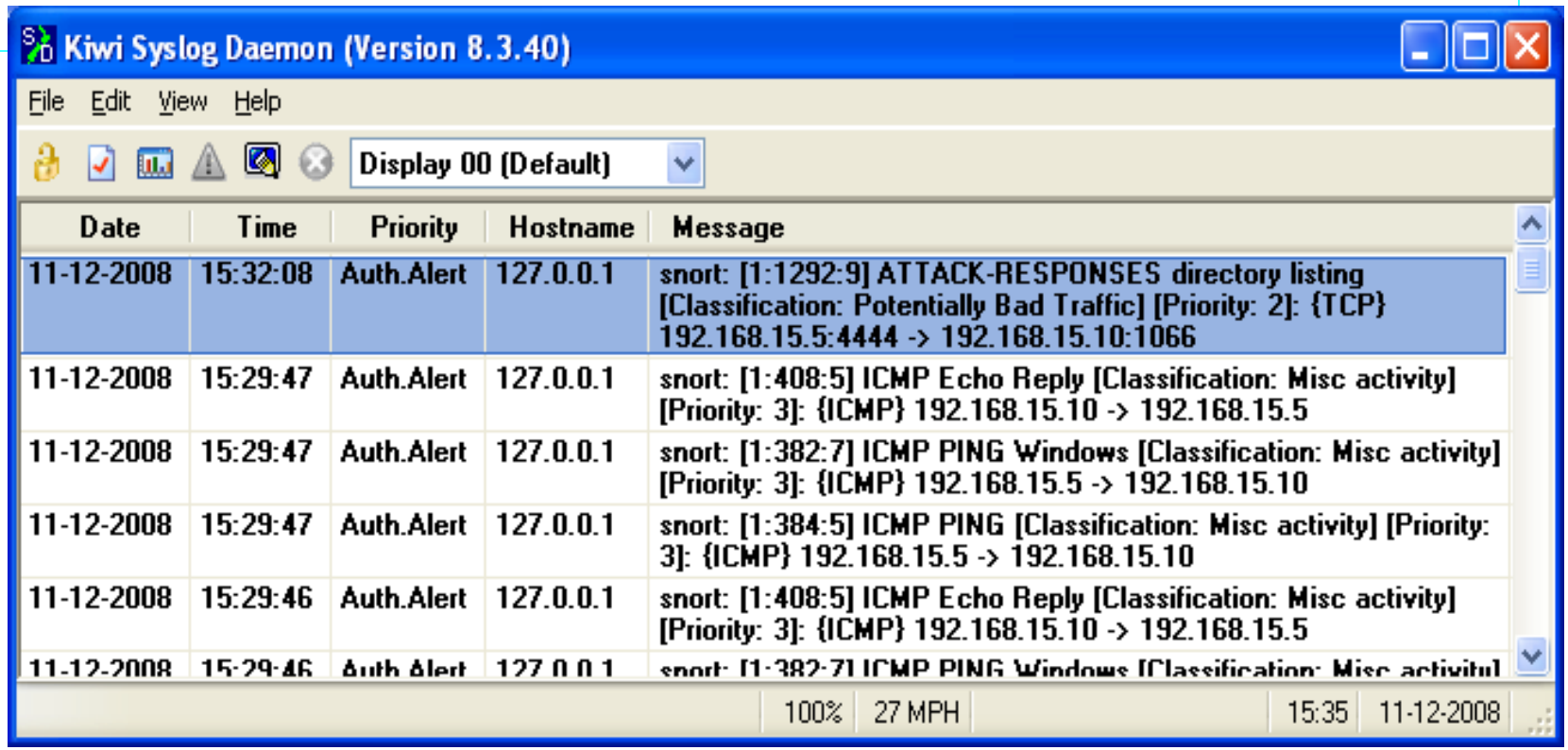
```
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any
(msg:"EXPLOITMicrosoft MMC createcab.cmd cross site scripting
attempt";flow:to_client,established;
content:"res|3A|//createcab.cmd";reference:bugtraq,19417;
reference:cve,2006-
3643;reference:url,www.microsoft.com/technet/security/bulletin/ms06-
044.mspx;classtype:attempted-user; sid:7424; rev:2;)
```

Rule body

- msg: prints a message in alerts and packet logs
- Flow rule option: used in conjunction with TCP stream reassembly
  - It allows rules to only apply to certain directions of the traffic flow
  - This allows rules to only apply to clients or servers
  - Established keyword will replace the TCP flags
- Content: search for a pattern in the packet's payload

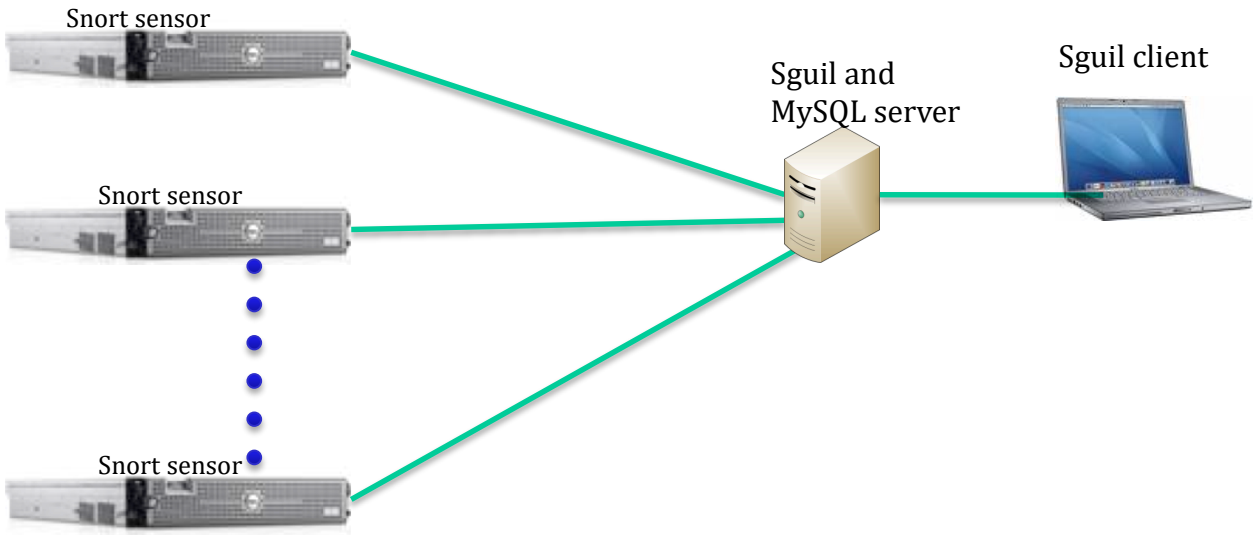
Snort report (1)

- Detect remote shell's DIR command

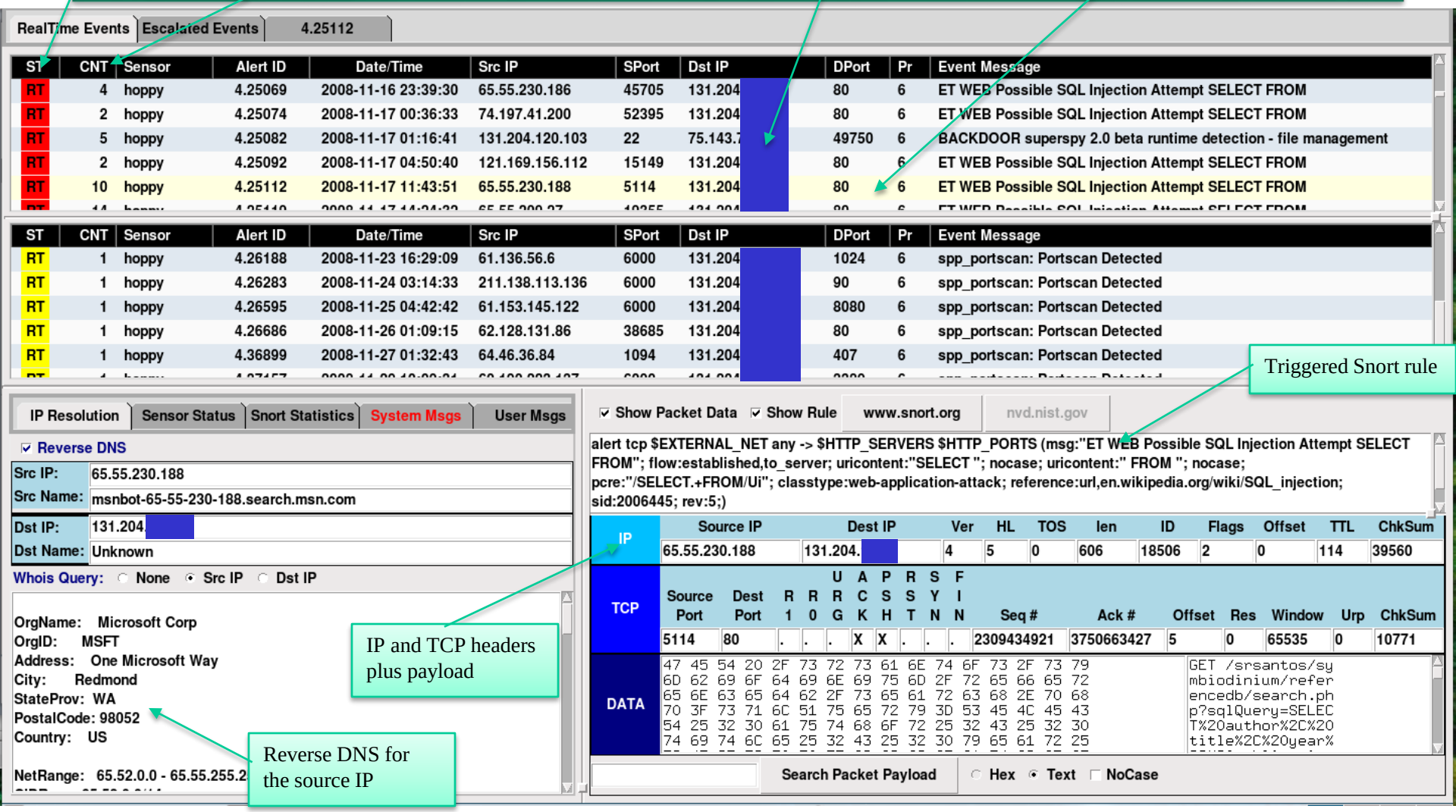


Sguil

- Sguil: Open Source Network Security Monitoring
  - http://nsmwiki.org/Main_Page
- The Sguil client is written in tcl/tk and can be run on any operating system that supports tcl/tk (including Linux, *BSD, Solaris, MacOS, and Win32)
- Snort alert and session data are stored in MySQL database
- Sguil server answers the query from Sguil client
- A good tutorial: Intrusion Detection FAQ: Build Securely Snort with Sguil Sensor Step-by-Step Powered by Slackware Linux, http://www.sans.org/resources/idfaq/slackware.php



Snort with Sguil



All correlated events

RealTime Events | Escalated Events | 4.25112

Close | Export

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	hoppy	4.25112	2008-11-17 11:43:51	65.55.230.188	5114	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25961	2008-11-21 08:55:27	65.55.230.188	3086	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25962	2008-11-21 08:55:30	65.55.230.188	3086	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25963	2008-11-21 08:55:36	65.55.230.188	3086	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25964	2008-11-21 08:55:48	65.55.230.188	3086	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25965	2008-11-21 08:55:57	65.55.230.188	3331	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25966	2008-11-21 08:56:00	65.55.230.188	3331	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25967	2008-11-21 08:56:06	65.55.230.188	3331	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.25968	2008-11-21 08:56:18	65.55.230.188	3331	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM
RT	1	hoppy	4.26111	2008-11-22 02:10:13	65.55.230.188	11067	131.204.	80	6	ET WEB Possible SQL Injection Attempt SELECT FROM

IP Resolution | Sensor Status | Snort Statistics | System Msgs | User Msgs

Reverse DNS

Src IP:	65.55.230.188
Src Name:	msnbot-65-55-230-188.search.msn.com
Dst IP:	131.204.
Dst Name:	Unknown

Whois Query: None | Src IP | Dst IP

OrgName: Microsoft Corp
OrgID: MSFT
Address: One Microsoft Way
City: Redmond
StateProv: WA
PostalCode: 98052
Country: US

NetRange: 65.52.0.0 - 65.55.255.255

Show Packet Data | Show Rule | www.snort.org | nvd.nist.gov

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"ET WEB Possible SQL Injection Attempt SELECT FROM"; flow:established,to_server; uricontent:"SELECT "; nocase; uricontent:" FROM "; nocase;

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
	65.55.230.188	131.204.	4	5	0	606	18506	2	0	114	39560

TCP	Source Port	Dest Port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	5114	80	.	.	.	X	X	.	.	.	2309434921	3750663427	5	0	65535	0	10771

DATA		
	47 45 54 20 2F 73 72 73 61 6E 74 6F 73 2F 73 79	GET /srsantos/sy
	6D 62 69 6F 64 69 6E 69 75 6D 2F 72 65 66 65 72	mbiodinium/refer
	65 6E 63 65 64 62 2F 73 65 61 72 63 68 2E 70 68	encedb/search.ph
	70 3F 73 71 6C 51 75 65 72 79 3D 53 45 4C 45 43	p?sqlQuery=SELEC
	54 25 32 30 61 75 74 68 6F 72 25 32 43 25 32 30	T%20author%2C%20
	74 69 74 6C 65 25 32 43 25 32 30 79 65 61 72 25	title%2C%20year%
	32 43 25 32 30 70 75 62 6C 69 63 61 74 69 6F 6E	2C%20publication
	25 32 43 25 32 30 76 6F 6C 75 6D 65 25 32 43 25	%2C%20volume%2C%
	32 30 70 61 67 65 73 25 32 30 46 52 4F 4D 25 32	20pages%20FROM%2

Search Packet Payload | Hex | Text | NoCase

Intrusion Detection Problems

- Polymorphic and metamorphic attacks use encryption and disguising techniques
  - Difficult to obtain signature and profile
- Lack of training data from real-world attacks
  - Difficult to capture the real attack data
  - Difficult to generalize all the possible evasions even when one case of an attack is captured
- 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
  - Administrator will spend many hours examining evidence

Internet Storm Center

- ISC was created in 2001
  - On March 22, 2001, intrusion detection sensors around the globe logged an increase in the number of probes to port 53
  - Within an hour of the first report, several analysts agreed that a global security incident was underway
  - They immediately sent a notice to a global community of security practitioners asking them to check their systems to see whether they had experienced an attack
  - Within three hours a system administrator in the Netherlands responded that some of his machines had been infected, and he sent the first copy of the worm code to the analysts
  - Just fourteen hours after the spike in port 53 traffic was first noticed, the analysts were able to send an alert to 200,000 people warning them of the attack in progress, telling them where to get the program to check their machines, and advising what to do to avoid the worm
  - The Li0n worm event demonstrated what the community acting together can do to respond to broad-based malicious attacks
  - Most importantly, it demonstrated the value of sharing intrusion detection logs in real time
- http://isc.sans.org/

IP Network Security Issues

- Eavesdropping
- Modification of packets in transit
- Spoofing (forged source IP addresses)
- Man-in-the-middle attack
- Denial of service
- Need for secure IP layer solution
  - SSL/TLS for Web using transport layer
  - S/MIME for email
  - SSH for remote login
- IPsec provides a open standards for secure communications over IP layer
  - Protect every protocol running on top of IPv4 and IPv6

IPsec: Network Layer Security

- Internet Key Exchange (IKE)
  - Authentication between two VPN parties
  - Establish security association for AH or ESP
  - Provide keys for AH or ESP
  - If IKE is broken, AH and ESP are not secure
- AH and ESP rely on an existing security association
  - Two parties must agree on
    - Crypto algorithms
    - A set of secret keys
    - IP addresses

IPsec = IKE + ESP + AH + Compression

Authentication + deriving keys for AH and ESP

Securing IP traffic
ESP: confidentiality + integrity
AH: integrity

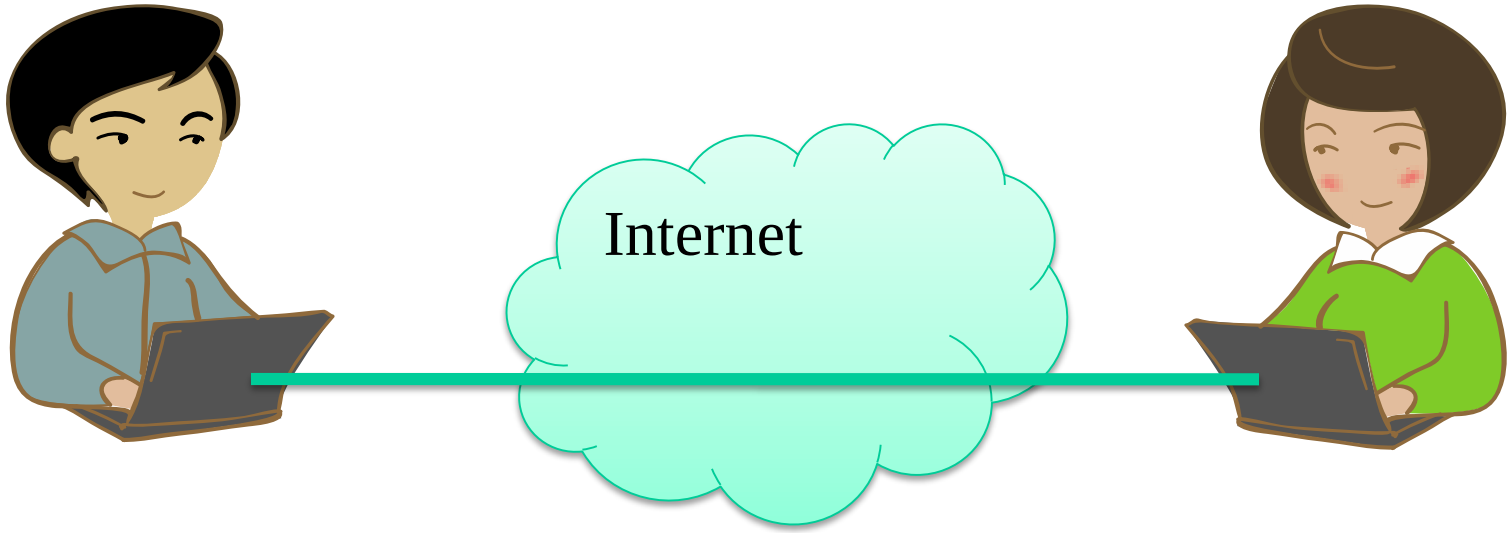
IPsec Security Services

- ESP and AH:
  - Authentication and integrity for packet sources
    - Connectionless integrity (for a single packet)
    - Partial sequence integrity (prevent packet replay)
- ESP:
  - Confidentiality (encapsulation) for packet contents
  - AES is supported
- Authentication and encapsulation can be used separately or together: However, encryption without authentication is not secure
- Both ESP and AH are provided in transport or tunnel mode
- These services are transparent to applications above transport (TCP/UDP/SCTP) layer

IPsec Modes

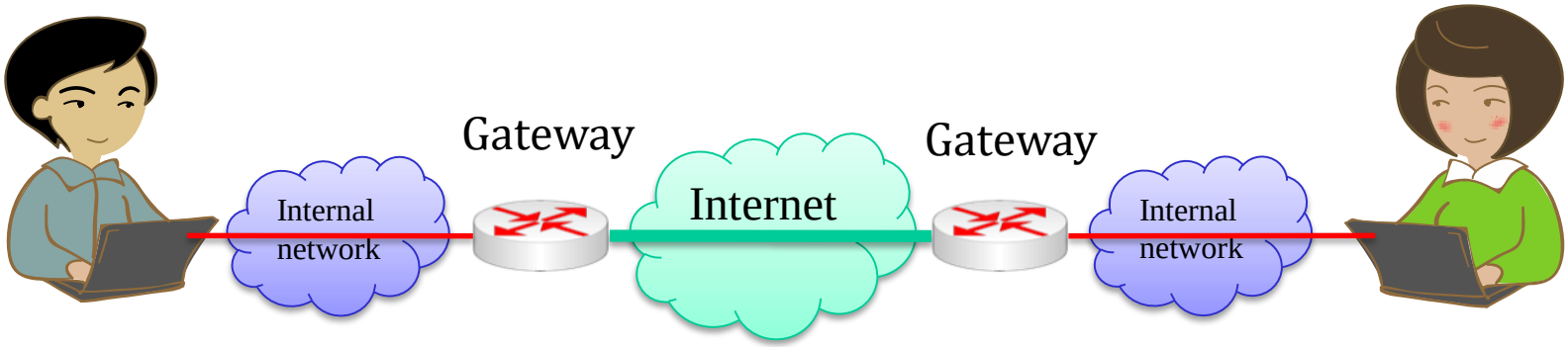
- Transport mode
  - Protection from
    - Host to host
    - Host to gateway
- Tunnel mode
  - Protection from
    - Gateway to gateway
      - Two gateways owned by the same organization
    - Host to gateway

IPsec in Transport Mode



- End-to-end security between two hosts
- Provide a secure channel across insecure networks (color: green)
- Both hosts need IPsec installed and configured

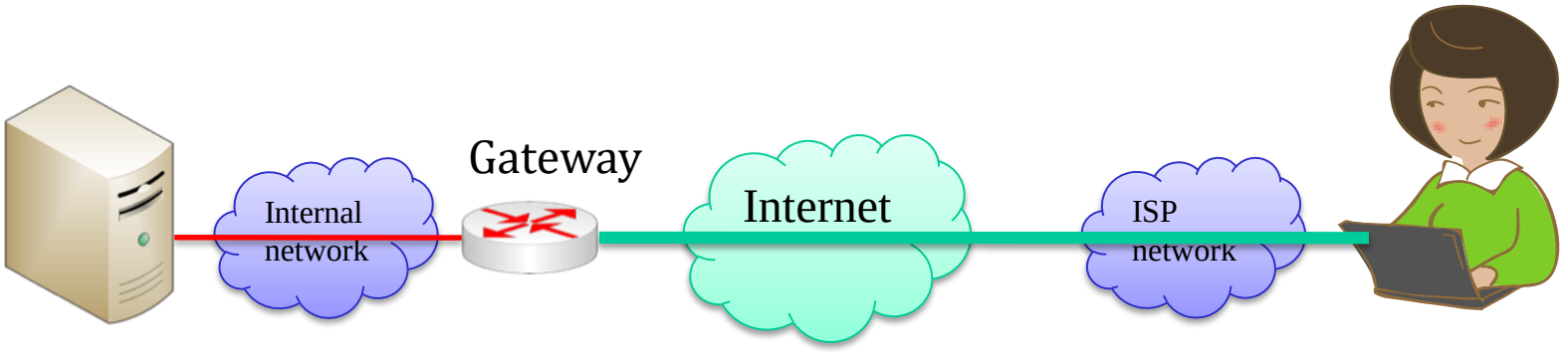
IPsec in Tunnel Mode



- Gateway-to-gateway security
  - Internal traffic inside gateway is not protected (color: red)
  - Virtual private network (VPN) across insecure Internet (color: green)
- Hosts do not need IPsec
- Gateways typically are routers configured with IPsec

Host to gateway

- Remote access to corporate network
  - Either tunnel or transport mode



Transport Mode vs. Tunnel Mode

- Transport mode
  - Protects packet payload
  - Uses original IP header

- Tunnel mode
  - Encapsulates both IP header and payload into IPsec payload

Security Association (SA)

- SA specifies how packets are protected
  - Cryptographic algorithms, keys, IVs, and lifetimes
  - Sequence numbers
  - Mode (transport or tunnel)
- One-way relationship between a pair of sender and recipient
  - Two SAs required for a two-way communication
- Each SA is uniquely identified by SPI (Security Parameters Index)
  - Each IPsec host keeps a database of SA's indexed by SPI
  - SPI is sent with packet so that recipient uses the SA to validate and extract information

Virtual Private Networks (VPN) tunnel

- ESP is often used to provide a VPN tunnel
  - Secure communication between two sites of the same organization over public unsecure Internet
  - Packets go from internal network to a gateway
    - IP headers contains source and destination IP addresses
  - Packets go from sending gateway to receiving gateway
    - Entire packet is hidden by encryption
    - Encryption Includes original headers so that source and destination IP addresses are hidden
    - The new IP header generated by the sending gateway indicates the source and destination IP addresses as the sending gateway and receiving gateway, respectively
  - Packets go from receiving gateway to receiving host
    - Gateway decrypts packet and forwards original IP packet to receiving host in the network that it protects

Network Address Translation (NAT)

- NAT problems
  - AH does not work with NAT
    - NAT must change information in the packet headers such as source IP address and source port number that are mapped by the NAT router
  - Encapsulating Security Payload (ESP) protocol:
    - Transport mode
      - If NAT is being used, one or both of the IP addresses are altered, so NAT needs to recalculate the TCP checksum
      - If ESP is encrypting packets, the TCP header is encrypted; NAT cannot recalculate the checksum, so NAT fails
      - TCP checksum calculation and verification is required in IPv4 whereas UDP can disable checksum in IPv4
      - UDP/TCP checksum calculation and verification is required in IPv6
    - Tunnel mode: compatible with NAT

SSL/TLS

- Secure Sockets Layer (SSL) protocol V3.0
  - De facto standard for web security
    - http://www.freesoft.org/CIE/Topics/ssl-draft/3-SPEC.HTM
- Transport Layer Security protocol, version 1.0 (RFC 2246)
  - Based on SSL V3.0
  - Current version TLS Version 1.2 : RFC 5246: “The Transport Layer Security (TLS) Protocol Version 1.2”
    - Same protocol design using different algorithms
- Supported by every Web browser and server
  - Protect information transmitted between browsers and Web servers
  - Provide privacy and data integrity between two communicating applications

SSL/TLS Protocol

- Contains Two protocols
- Handshake protocol
  - Use public-key cryptography to establish a shared secret key between the client and the server
- Record protocol
  - Use the secret key established in the handshake protocol to protect communication between the client and the server

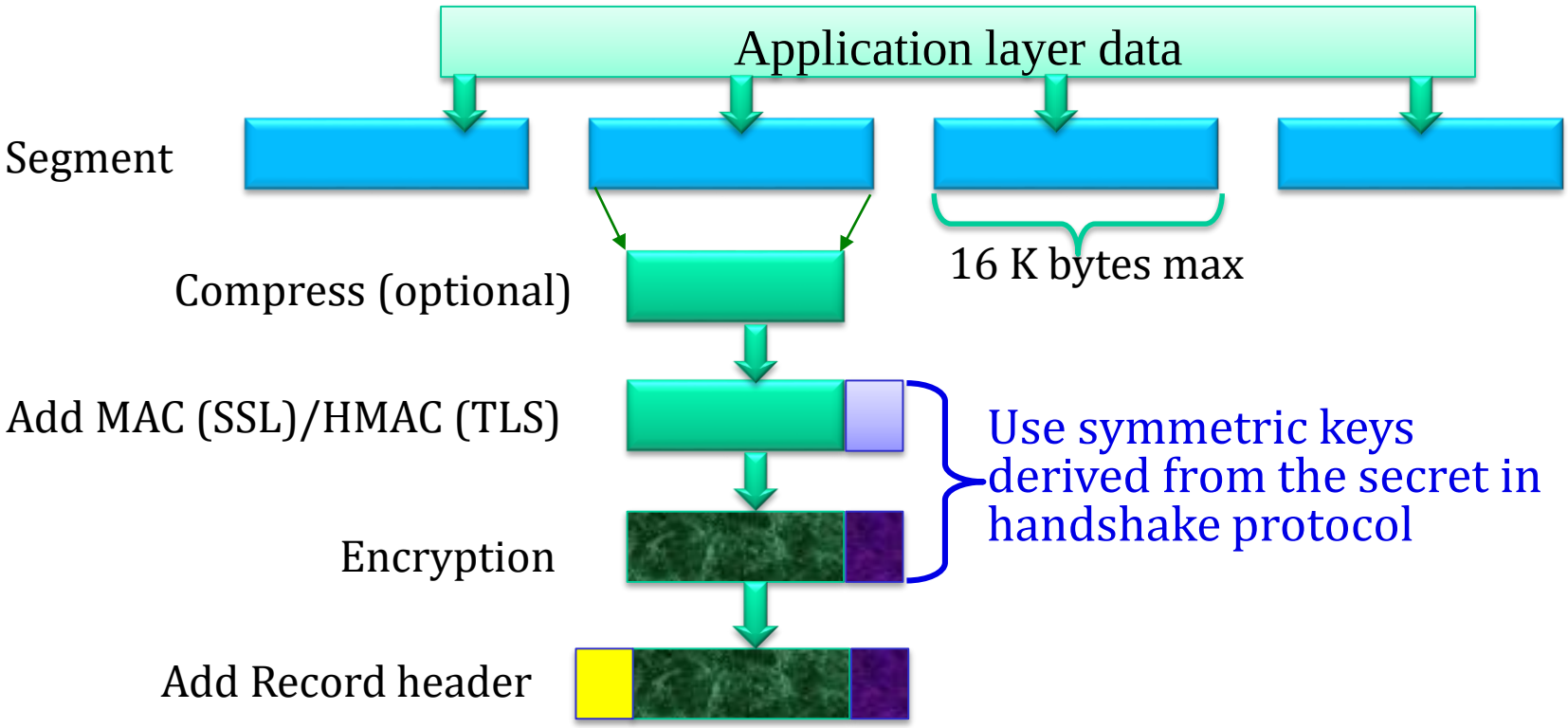
SSL/TLS Handshake Protocol

- Negotiate version of the protocol and the set of cryptographic algorithms to be used
  - Interoperability for different implementations
- Two parties: client and server
  - Authenticate server using certificate
  - Optionally authenticate client
    - Use client certificate or password
    - RFC 4279: Pre-Shared Key Ciphersuites for Transport Layer Security
    - RFC 5054: Using the Secure Remote Password (SRP) Protocol for TLS Authentication
- Use public key to establish a shared secret for symmetrical crypto in record protocol
  - Symmetrical crypto: AES, or 3DES
  - For encrypting credit card number/communication

Hello and Key Exchange

- The client hello and server hello establish the following attributes:
  - Protocol version, session ID, cipher suite, and compression method
  - Two nonces are generated and exchanged:
    - ClientHello.random and ServerHello.random
- The client key exchange message
  - The content of that message will depend on the public key algorithm selected between the client hello and the server hello
    - If the client has sent a certificate with signing ability, a digitally-signed certificate verifies the message is sent to explicitly verify the certificate
      - Prove the private key
- The server key exchange message sent by the server if it
  - Has no certificate
  - Has a certificate that is only used for signing
    - (e.g., DSS certificates, signing-only RSA certificates)
    - This message is not used if the server certificate contains Diffie-Hellman parameters

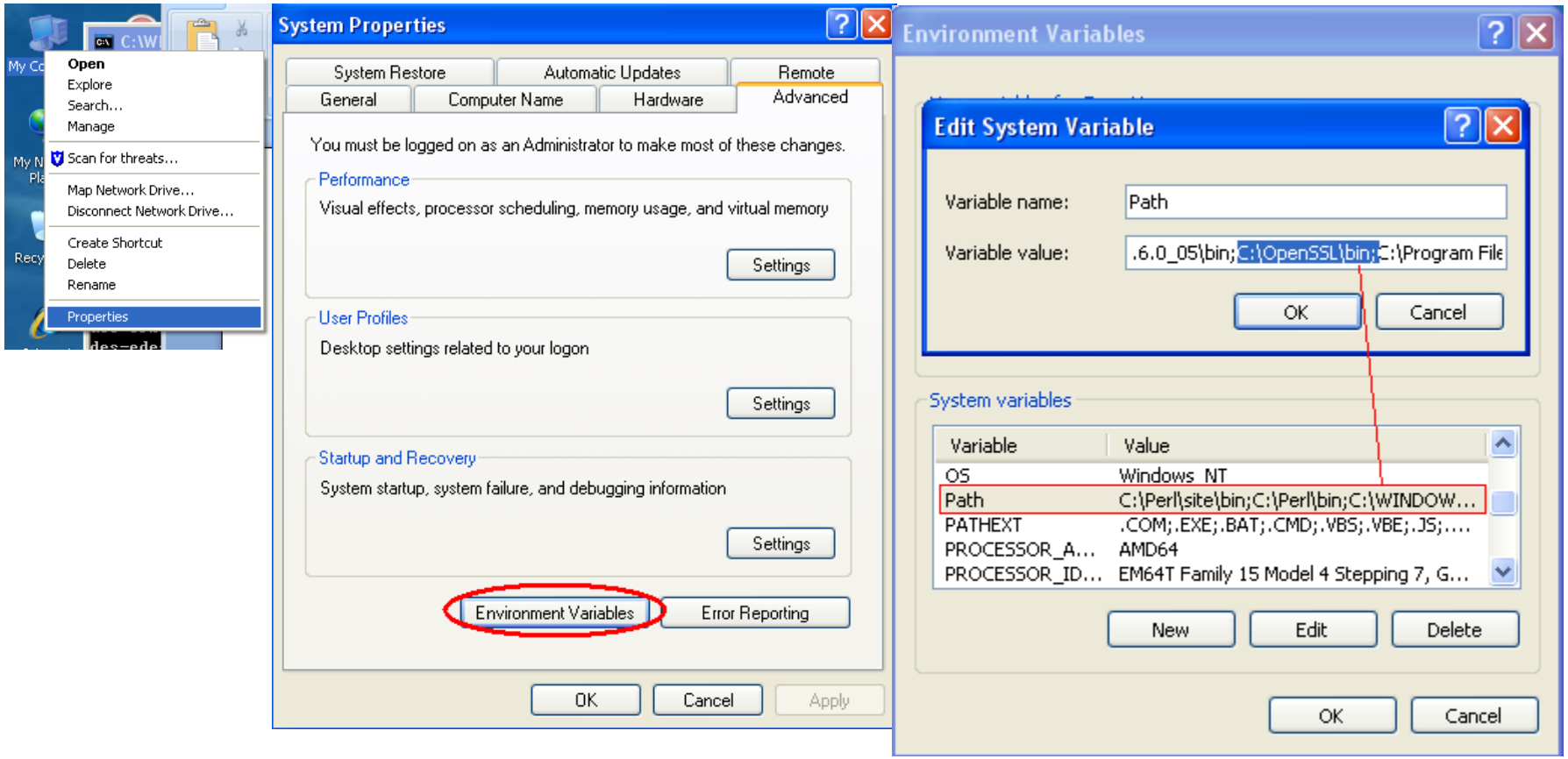
SSL/TLS Record Protocol Packet



Create your own CA

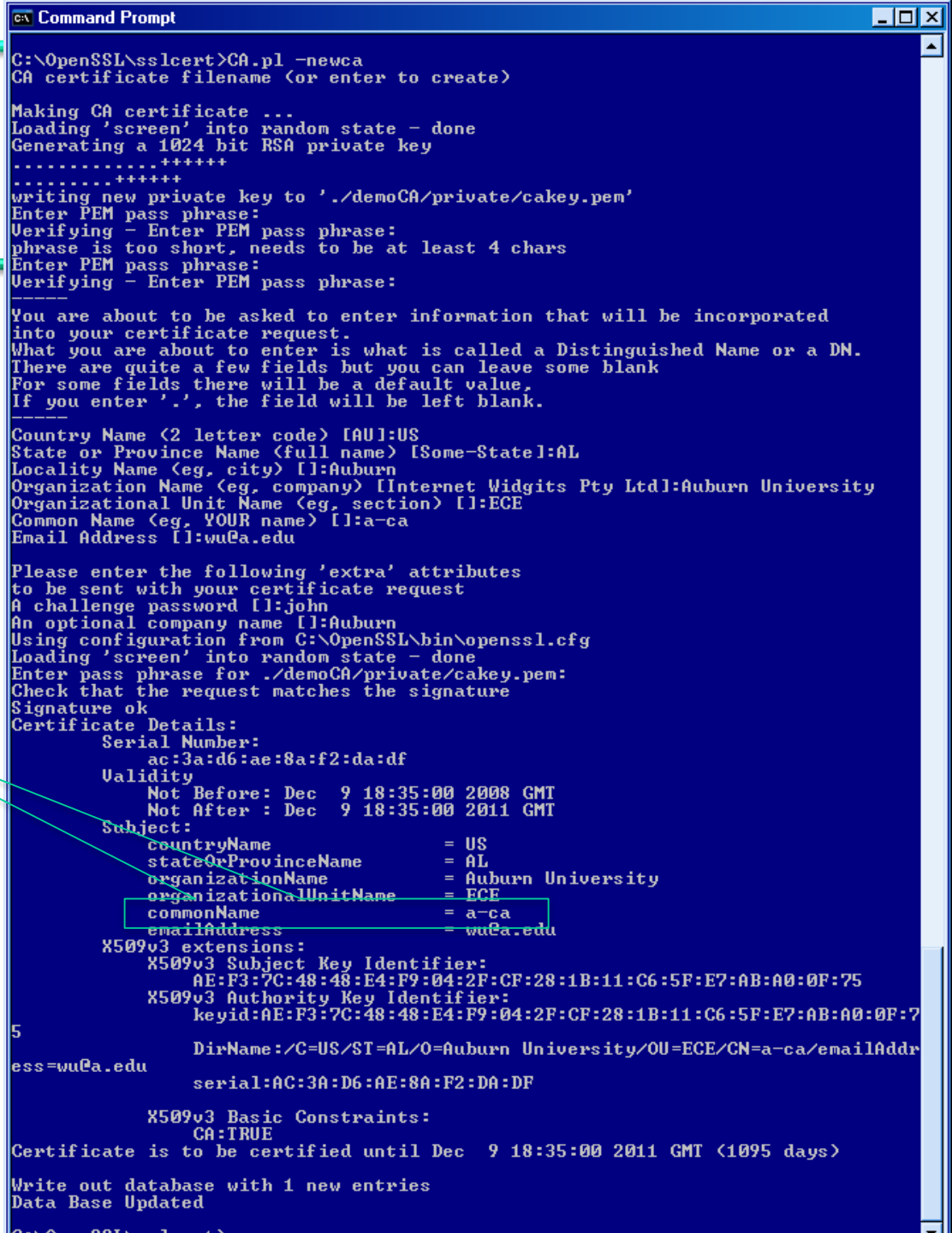
- OpenSSL
  - An open source implementation of the SSL and TLS protocols
  - The core library (written in the C programming language) implements the basic cryptographic functions and provides various utility functions
  - Wrappers allowing the use of the OpenSSL library in a variety of computer languages are available
  - Available for any OS
- Install procedure in Windows
  - Install the free ActivePerl
    - http://activestate.com/Products/activeperl/
  - Install OpenSSL
    - http://www.slproweb.com/products/Win32OpenSSL.html
    - Install Visual C++
    - Install OpenSSL is: C:\OpenSSL
  - Modify PATH variable in Windows

Add C:\OpenSSL\bin in Path



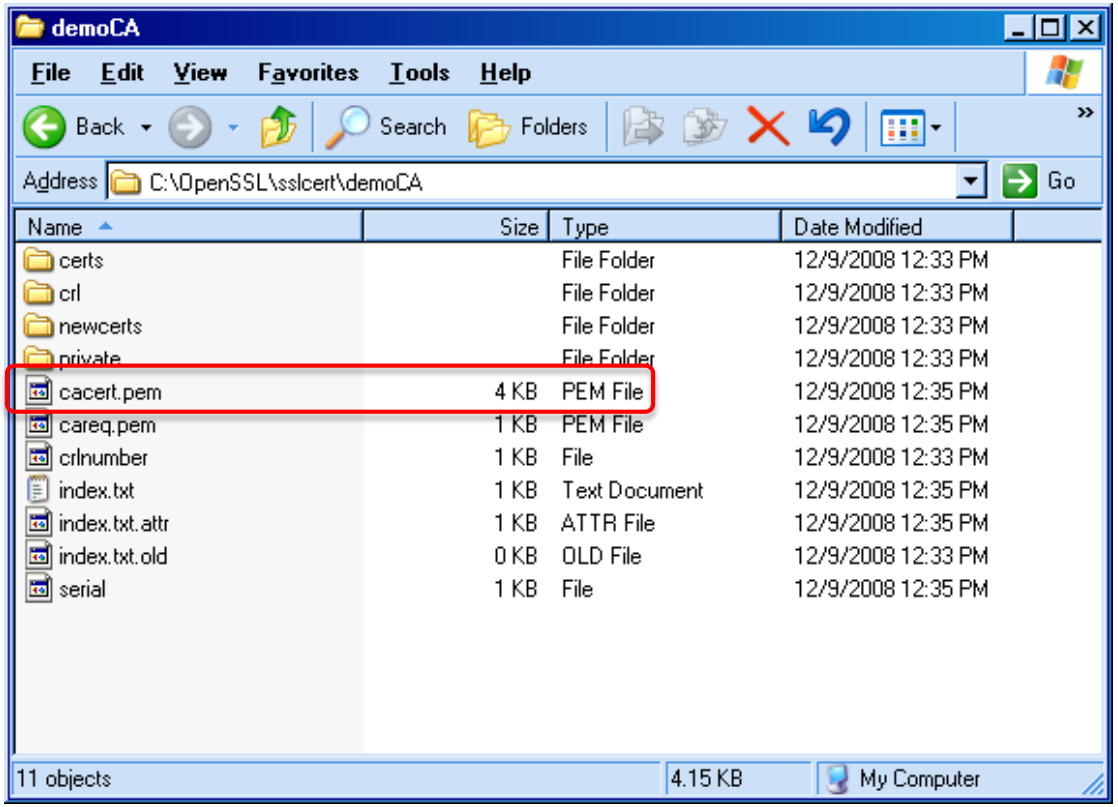
Create CA

- Create self-signed certificate for CA
  - Generate a keypair that is exportable and a self-signed certificate as a trusted root certificate
  - CA's name: a-ca
  - CA root certificate: c:/openssl/sslcert/demoCA/CAcert.pem



CA certificate

- CA certificate



- OpenSSL generates keys and certificates in PEM file format
- Unfortunately der format is the one that can be accepted by Apache and Windows
- To convert a certificate from PEM to der:

Openssl x509 -in cacert.pem -inform PEM -out cacert.crt -outform der

Generate a web site's Certificate

- IIS:
  - First generate a Certificate Signing Request (CSR) by IIS in the web server, and save the request in the C:\ directory called certreq.txt. The request format should be changed to newreq.pem to match the OpenSSL format
- Apache:
  - Use OpenSSL to create a key pair first in order to generate a Certificate Signing Request (CSR)

```
openssl genrsa -aes128 -out a-edu.key 1024
```

  - Generate a certificate request with the RSA private key (output will be PEM format):

```
openssl req -new -key a-edu.key -out a-edu.pem
```

  - This will generate a request file a-edu.pem

The Steps for Generating a Certificate Signing Request (CSR) for a domain

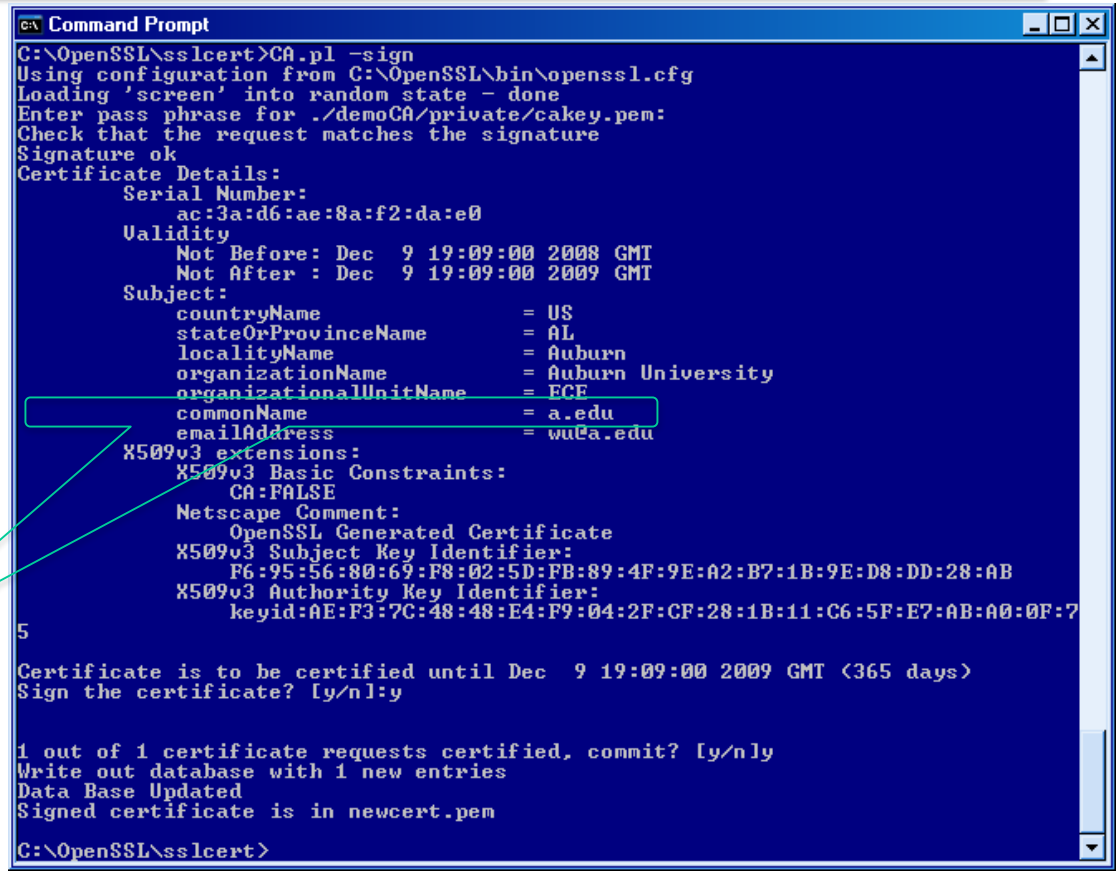
```
C:\Users\wu\wu_ca>openssl req -new -key a-edu.key -out a-edu.csr
Enter pass phrase for a-edu.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:AL
Locality Name (eg, city) []:Auburn
Organization Name (eg, company) [Internet Widgits Pty Ltd]:auburn.edu
Organizational Unit Name (eg, section) []:ECE
Common Name (eg, YOUR name) []:a.edu
Email Address []:wu@auburn.edu

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:YourPassword
An optional company name []:a.edu
```

A Certificate Request


```
-----BEGIN CERTIFICATE REQUEST-----
MIIDFzCCAf8CAQAwgZExCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJBTDEPMA0GA1
UEBxMGQXVidXJuMRMwEQYDVQQKEwphdWJ1cm4uZWR1MQwwCgYDVQQLEwNFQ0Ux
HzAdBgNVBAMTFlcyMDA4UjIudWF2LmF1YnVybi5lZHUxIDAeBgkqhkiG9w0BCQ
EWEXd1QGVuZy5hdWJ1cm4uZWR1MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAt77e6B9GC5NJFpMxn6F2hetSR5fggn+h3u5c2ffDJnxS86lYE+yEKX
BzwQkgU7ySweqxHqqjkCdLJuH+GN08o2yQPnt7LyTc6Ry1M0JHbaSpN3hjY+df
KVKgbmKEGmGTDQ+3G2Ju2NKGxGYNsgWHIApsNm+e++kFah6Nas+p+q1deZyIqA
Y5gpcleBjgTBi8A498i4VfLncFObr3cKXHVD9/aD85IyWdYOXifoKzbpGv89Zp
XafzVSBnA1mkLBXpxTHWFnFNmK9KLzYo0Uo4ljSUML0dTz+9ntMcs/psrk3puc
e6cG07Xb8EiGGJnct3EKLkHz5N0sBSLdP8+ZZNPwIDAQABoEAwFwYJKoZIhvcN
AQkHMQoTCEppYWphbi1zMCUGCSqGSIb3DQEJAjEYExZXMjAwOFIyLnVhdi5hdW
J1cm4uZWR1MA0GCSqGSIb3DQEBBQUAA4IBAQBDik4iTGRZC2XbeT2YPkWu5XD2
ka0kIFFZ7mhNkK3O1yxM8PBXpqWMQujlxNCE2aX4N7w5kPqZCb9vwXccrfHsmS
NytEkcCn9nlnPjIg3Bei/PlGzo8jWSa+jJmZ+7wxwxhQzGXvLauoabLg1qcSMY
DRnjh+kcRiwmBX5PNNaqaYfoMcsO5DKkz4QESGy8NtV2s1OEyNfTA+cNwsIxW1
vu0OFV/5SrpAbi0YKgdpJ7DaUHisIPBEimYvamTu0kQVIQOctXqLyUmeoSPUJh
4SVZs6Ds1oNPGYeicFvJOwlYzFUQXLgZ0lQLoshADIFk0Mdm4iZXBYobQ3yl5B
nU0SSA-----END CERTIFICATE REQUEST-----
```

Sign a certificate request

- Rename the request file from a-edu.pem to newreq.pem
- Execute the CA -sign command which signs a request using the private key of the root CA, held in private/cakey.pem.
  - The request needs to be in a file called newreq.pem
  - The generated certificate is written to a file called newcert.pem
  - The website's name is a.edu



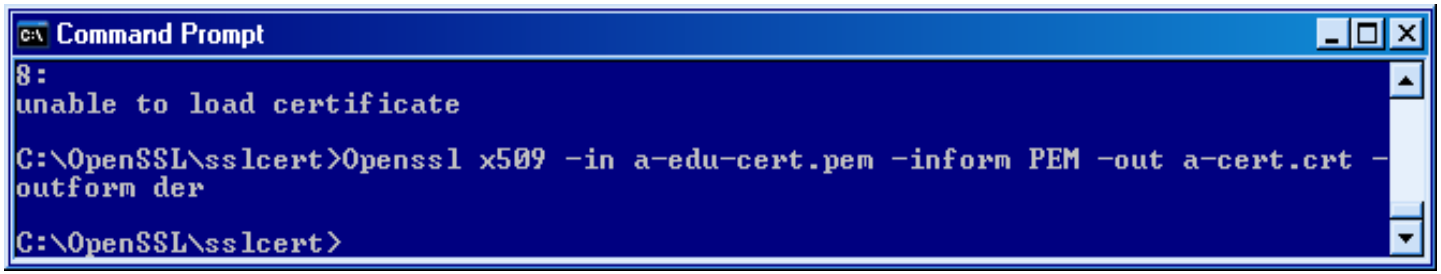
File format conversion

- To convert a certificate from PEM to der:

Openssl x509 -in a-edu-cert.pem -inform PEM -out a-cert.crt -outform der

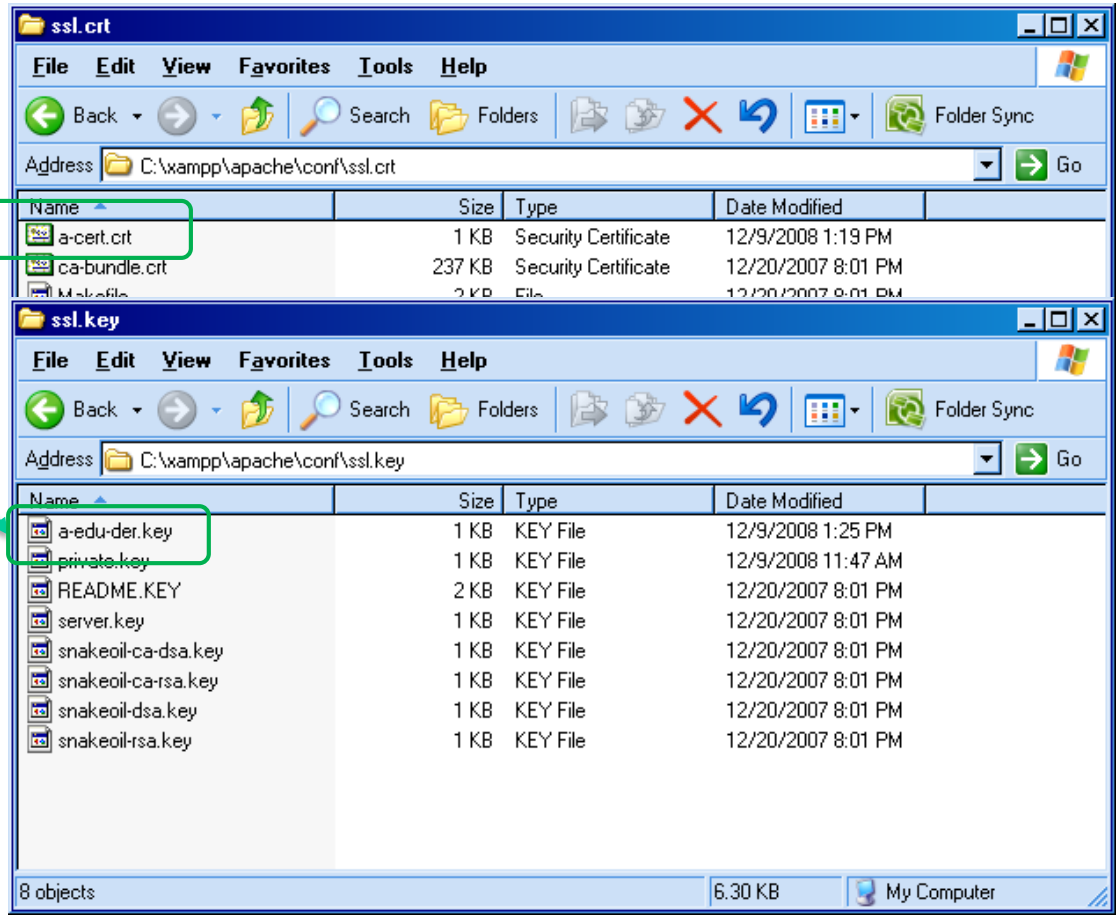
- To convert a key from PEM to DER:

Openssl rsa -in a-edu.key -inform PEM -out a-edu-der.key -outform DER



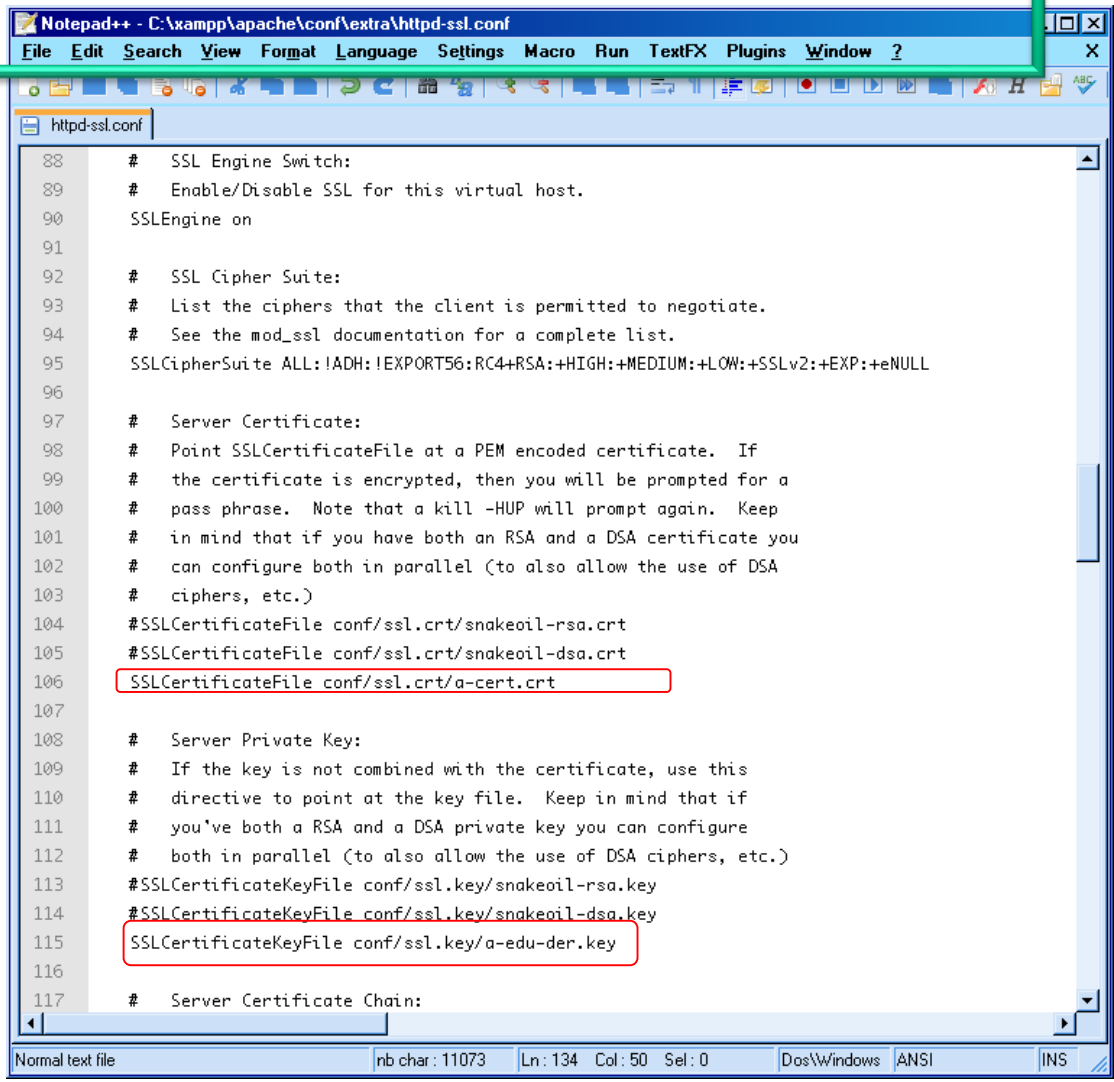
Put certificate and key in the right folders for Apache

- Certificate file
- Key file

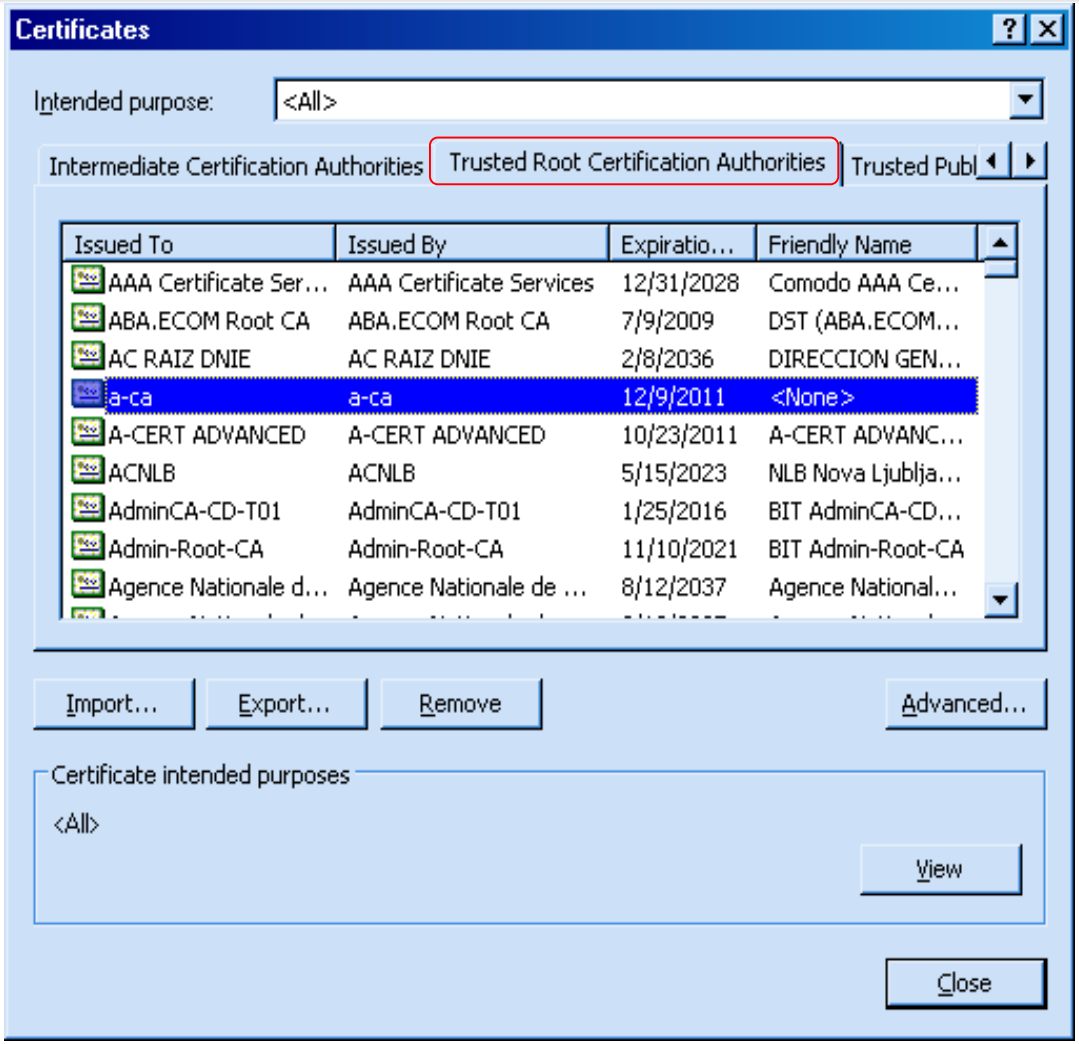


Configure Apache

- Specify the certificate and private key files



Verify the root CA certificate is in place



Test Apache for https