

Security & Privacy in Computing

CS 600.443

Professor Avi Rubin

(Introduce myself)

The course

- Assignments:
 - Groups of 1-4 students
 - Interactive: groups will present to class at various phases
- Lecture topics in Security & Privacy
- Grading:
 - Assignments: 50%
 - Midterm 20% Final: 30%
- Two tracks: implementation heavy or
 write papers + midterm

Lecture topics

- Network security
- Cryptography
- Wireless security
- Voting systems and requirements
- Web Security
- Viruses & Worms
- Defense mechanisms
- Authentication & Passwords
- Secure Programming, guest lecturer, Gary McGraw
- Host-level security
- Safe tools
- Privacy
- P3P
- Anonymity technologies

Some administrative details

- Policies:
 - most lecture slides will not be made available (take notes)
 - late assignments penalized 5 points/day late
 - Collaboration only allowed on assignments. Must take your own exams. Exams are closed book and closed notes.
- Course text:
 - Firewalls & Internet Security 2e
- Course web page:
 - <http://www.cs.jhu.edu/~rubin/courses/sp04/syllabus.html>

More administration

- Class mailing list
 - send mail to `majordomo@cs.jhu.edu` with:
subscribe cs443
in message body.
- TA: Matt Green `mgreen@cs.jhu.edu`
 - also helping: Chris Soghoian
- Office hours: after class Thursdays in 326 NEB
 - (except today)
- During class, please keep cell phones on vibrate or turn them off
- Current events: please bring relevant news stories to my attention (and be prepared to discuss in class). I like to occasionally digress from the syllabus for major events
- Go through syllabus
- Need to end a few minutes early today
 - time to meet students and start forming groups

Survey - show of hands

- Programmed w/a Crypto API
- Configured a Firewall for > 10 people
- Configured a personal firewall
- Know how to read tcpdump output
- Have run nmap or snort
- Understand how a buffer overflow works
- Read a privacy policy
- Written a privacy policy
- Hacked into someone else's system
- Written a virus or worm

Registration status?

Questions?

Network Threats

Network mapping

- Attacker probes a network to learn
 - Topology/architecture
 - Discover what hosts exists
 - Discover services running on hosts
 - Expose vulnerabilities
- ICMP echo
- Port scanning
 - Send TCP SYN and wait for SYN/ACK
 - E.g. nmap and nessus

Defense against mapping

- Application-level gateways
- Program: `iplog` sends bogus responses to `nmap` queries
- Monitor network with program like Network Flight Recorder (NFR)
- Use a fingerprint scrubber, which defeats TCP/IP stack fingerprinting
 - Different TCP/IP implementations react differently to certain probes, used by attackers to map hosts

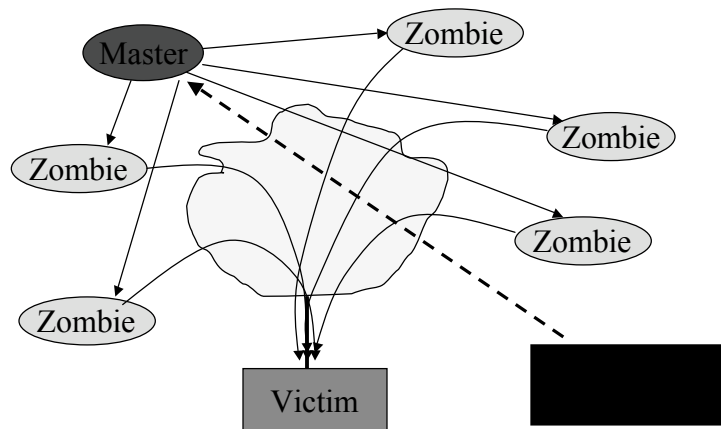
Denial of service

- Attacks against availability of resources
- Can sometimes be accidental
- Many automated tools exist for DOS
- Easy to detect attack (by definition)
- Difficult to prevent attacks
 - Multiple instances of legitimate network traffic
- Latest threat:
 - Distributed Denial of Service (DDOS)

What Are DDoS Tools?

- Clog victim's network.
- Use many sources (“zombies”) for attacking traffic.
- Use “master” machines to control the daemon attackers.
- At least 4 different versions in use: TFN, TFN2K, Trinoo, Stacheldraht.

How They Work



How They Talk

- Trinoo: attacker uses TCP; masters and zombies use UDP; password authentication.
- TFN: attacker uses shell to invoke master; masters and daemons use ICMP ECHOREPLY.
- Stacheldraht: attacker uses encrypted TCP connection to master; masters and daemons use TCP and ICMP ECHO REPLY; rcp used for auto-update.

Deploying DDOS

- Attackers seem to use standard, well-known holes (i.e., rpc.ttdbserver, amd, rpc.cmsd, rpc.mountd, rpc.statd).
- They appear to have “auto-hack” tools – point, click, and invade.
- Lesson: practice good computer hygiene.

Detecting DDOS Tools

- Most current IDS's detect the current generation of tools.
- They work by looking for DDOS control messages.
- Naturally, these will change over time; in particular, more such messages will be properly encrypted.

What are the Strong Defenses?

- Still in the research phase...

What Can ISPs Do?

- Deploy source address anti-spoof filters (*very important!*).
- Develop security relationships with neighbor ISPs.
- Set up mechanism for handling customer security complaints.
- Develop traffic volume monitoring techniques.
 - Look for too much traffic to a particular destination.
 - Learn to look for traffic to that destination at border routers (access routers, peers, exchange points, etc.).

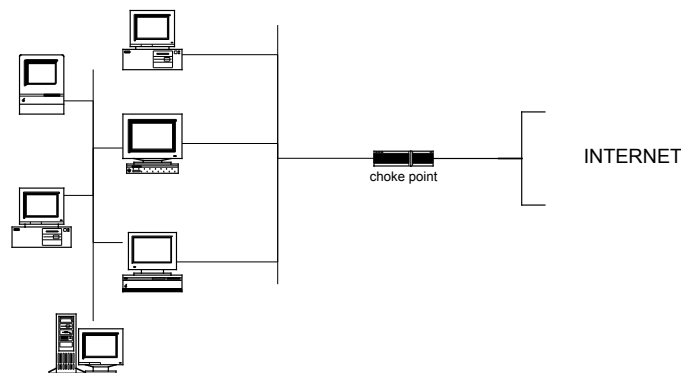
Enhanced Congestion Control

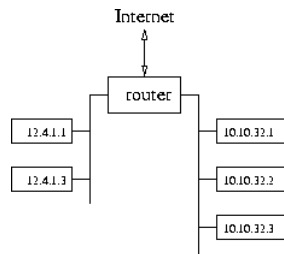
- Define an attack as “too many packets drops on a particular access line”.
- Send upstream node a message telling it to drop more packets for this destination.
- Issues: authentication, fairness, effect on legitimate traffic, implementability, etc.

Firewalls

- Protecting a perimeter
 - Only works against *outsiders*
 - Choke point: force all traffic through one node
 - It can be tested
 - Policy decision can be made
- Packet filters
 - Look at TCP and IP headers
- Application level gateways
 - Proxies for services, usually require client rewrite

Choke point





- Hosts on the right use private addresses
- To compromise them from Internet
 - must first compromise host on the left

Configuring firewall

- Must know what should be permitted and denied
 - have a policy
- Allowable packets must be formally specified
 - logical expressions on packet fields
- Expression must be rewritten in vendor syntax

Rules

action	ourhost	port	theirhost	port	comment
block	*	*	SPIGOT	*	<i>we don't trust these people</i>
allow	OUR-GW	25	*	*	<i>connection to our SMTP port</i>

- Rules:
 - block all packets to host spigot
 - allow packets to our GW on smtp port

action	ourhost	port	theirhost	port	comment
allow	*	*	*	25	<i>connection to their SMTP port</i>

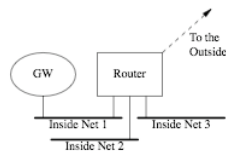
- Allow packets from port 25
 - problem: attacker can originate packets from port 25 and reach any port.

More rules

action	src	port	dest	port	flags	comment
allow	{our hosts}	*	*	25		<i>our packets to their SMTP port</i>
allow	*	25	*	*	ACK	<i>their replies</i>

- Allow packets from our host to their SMPT port
- Allow replies
- Define rules for each interface

Example



- limited connections between GW and outside
- limited connection from GW to net2 or net3
 - why?
- anything can pass between net2 and net3
- outgoing only between net2 & net3 and outside

External interface rule set

action	src	port	dest	port	flags	comment
block	{NET 1}	*	*	*		<i>block forgeries</i>
block	{NET 2}	*	*	*		
block	{NET 3}	*	*	*		
allow	*	*	GW	25		<i>legal calls to us</i>
allow	*	*	{NET 2}	*	ACK	<i>replies to our calls</i>
allow	*	*	{NET 3}	*	ACK	

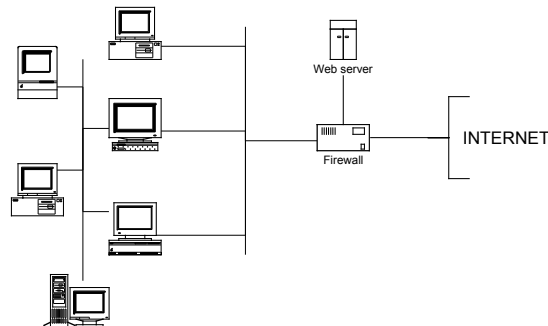
- no packets can originate from the inside on the outside interface
- allow calls to the GW (mail)
- allow replies to existing connections

Net 1 interface

action	src	port	dest	port	flags	comment
allow	GW	*	{partners}	25		mail relay
allow	GW	*	{NET 2}	*	ACK	replies to inside calls
allow	GW	*	{NET 3}	*	ACK	
block	GW	*	{NET 2}	*		stop other calls from GW
block	GW	*	{NET 3}	*		
allow	GW	*	*	*		let GW call the world

- Allow mail relaying
- Allow replies to existing connections
- don't allow GW to connect to net2 or net3
- Do not block the GW from the outside

Web server placement



Distributed Firewalls

- Individual hosts enforce security policy
- Rule created by admin
 - shipped out to every machine
 - over an authenticated channel
- Advantages
 - no central point of failure (sort of)
 - When used w/cryptography, can allow flexible policies
- Disadvantages?

What firewalls cannot do

- block infected floppy disk w/virus
- limited scanning of email possible
- can't solve people problems w/software
- scan FTP up/down loads (too much work)
- transitive trust
 - A trust B to let traffic in, and B trusts C, then A trusts C, whether he likes it or not
- Many firewalls have errors, or do not work as expected
 - configuration errors
 - bugs in the firewall itself

Creating a firewall policy

- Insiders are trusted:
 - initiate outgoing TCP connections
 - run ping and traceroute
 - issue DNS queries
 - set clock using an external time server
- Outside world cannot initiate connections in
- Access to mail and other services done by Polling

Personal firewalls

- For Windows:
 - Symantec Internet security product
 - ZoneAlarm (purchased by checkpoint)
- Identify applications on a machine
 - control what each application can do
- Provide alerts (at times too many)
- Quite different from packet filters
- Available on many different platforms
 - e.g. brickhouse (GUI for apple fw)

Building a firewall w/ipchains

- chains: set of rules that logically fit together
- Input
 - decisions when packet enters an interface
- output
 - decisions when packets leave an interface
- forward
 - used to make routing decisions or *masquerading*
- Chains reside in kernel and are loaded at startup

IPchains

- There are user-defined chains and system chains
 - system chains load at startup
 - User chains can be assigned a name, to group by logical order
- IPchains used to protect a single computer

IPchains examples

- `ipchains -A input -j ACCEPT -p TCP -s 135.207.10.208`
 - allow outbound TCP traffic
- `ipchains -A input -j ACCEPT -p TCP ! -y -d 135.207.10.208`
 - allow inbound traffic, except if SYN flag is set

```
ipchains -A input -j ACCEPT -p UDP -d 135.207.10.208 -s 0/0 domain
```

```
ipchains -A input -j ACCEPT -p UDP -s 135.207.10.208 -d 0/0 domain
```

```
ipchains -A input -j ACCEPT -p UDP -d 135.207.10.208 -s 0/0 ntp
```

```
ipchains -A input -j ACCEPT -p UDP -s 135.207.10.208 -d 0/0 ntp
```

- allow DNS traffic and NTP traffic, no other UDP traffic is allowed
- default rule at end disallows everything

Sample policy

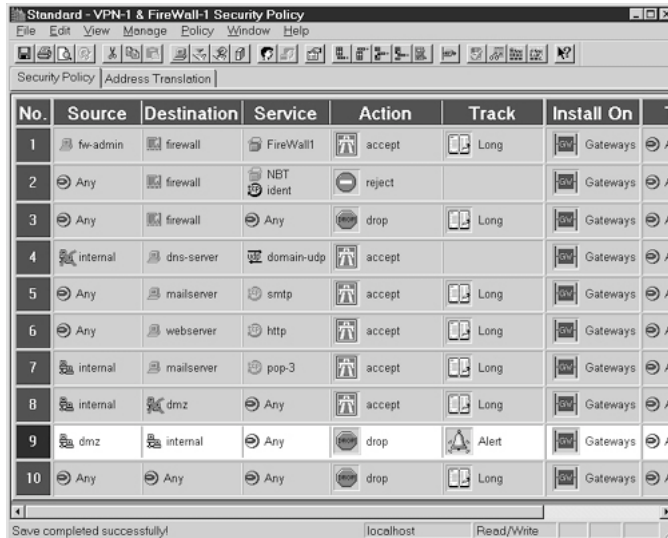
Chain input (policy ACCEPT):

target	prot	opt	source	destination	ports
ACCEPT	tcp	-y----	rubinlap	anywhere	any -> any
ACCEPT	tcp	-----	rubinlap	anywhere	any -> any
ACCEPT	tcp	!y----	anywhere	rubinlap	any -> any
ACCEPT	udp	-----	anywhere	rubinlap	domain -> any
ACCEPT	udp	-----	rubinlap	anywhere	any -> domain
ACCEPT	udp	-----	anywhere	rubinlap	ntp -> any
ACCEPT	udp	-----	rubinlap	anywhere	any -> ntp
ACCEPT	icmp	-----	rubinlap	anywhere	echo-request
ACCEPT	icmp	-----	rubinlap	anywhere	echo-reply
ACCEPT	icmp	-----	anywhere	rubinlap	echo-request
ACCEPT	icmp	-----	anywhere	rubinlap	echo-reply
ACCEPT	icmp	-----	anywhere	rubinlap	time-exceeded
ACCEPT	icmp	-----	anywhere	rubinlap	fragmentation-needed
ACCEPT	tcp	-y----	anywhere	rubinlap	any -> auth
DENY	all	---I-	anywhere	anywhere	n/a

Chain forward (policy DENY):

Chain output (policy ACCEPT):

Firewall rules



The screenshot shows a window titled "Standard - VPN-1 & FireWall-1 Security Policy". The window contains a table with 10 rows of firewall rules. The columns are: No., Source, Destination, Service, Action, Track, Install On, and T. The rules are as follows:

No.	Source	Destination	Service	Action	Track	Install On	T
1	fw-admin	firewall	FireWall1	accept	Long	Gateways	A
2	Any	firewall	NBT Ident	reject		Gateways	A
3	Any	firewall	Any	drop	Long	Gateways	A
4	internal	dns-server	domain-udp	accept		Gateways	A
5	Any	mailserver	smtp	accept	Long	Gateways	A
6	Any	webserver	http	accept	Long	Gateways	A
7	internal	mailserver	pop-3	accept	Long	Gateways	A
8	internal	dmz	Any	accept	Long	Gateways	A
9	dmz	internal	Any	drop	Alert	Gateways	A
10	Any	Any	Any	drop	Long	Gateways	A

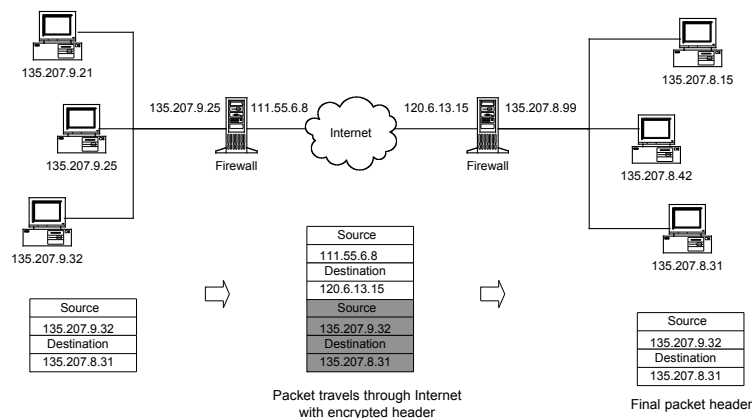
First assignment

- Design the network architecture
 - mail server, web server, high compute client, regular client, router(s): 3+ separate nets
 - identify all interfaces
- For each interface, specify firewall policy
- Using IPchains on Linux or BSD, define policies for each interface, and print out the IPchains listing

Virtual Private Networks

- VPN
- Enable distant computers to share the same virtual security perimeter
- Can be
 - Site to site
 - Laptop to site
 - Laptop to laptop
- Typically utilize IPsec, security at the IP layer

VPN example



Intrusion Detection

- Software to monitor networks
 - Looks for anomalies
 - False alarm rates are typically high
- IDS good for
 - Knowing what is going on in the network
 - Post-processing after an attack
 - Early warning system
- IDS not so great for
 - Detecting new types of attacks
- IMHO IDS is a bit overrated

Evaluating security

- Evaluate
 - Network
 - Conformance to site policy
 - Implementation of policy, e.g. firewall rules, pw rules, etc
 - IDS?
 - Systems in use
 - Keep current on bug updates, patches, alerts
 - Make sure it is most secure (e.g. story with SSH v1 vs. v2)
 - Code
 - Are programmers security literate?
 - Use of APIs and packages (use right packages, correct usage)
 - Choice of parameters
 - Choice of programming languages
 - Code review

Testing security

- Penetration testing
- Port mapping
 - Don't be the second person to run nmap, nessus on your system
- Cracking your users' passwords, l0phtcrack
- Test firewall rules, e.g. Lumeta analyzer

BGP security

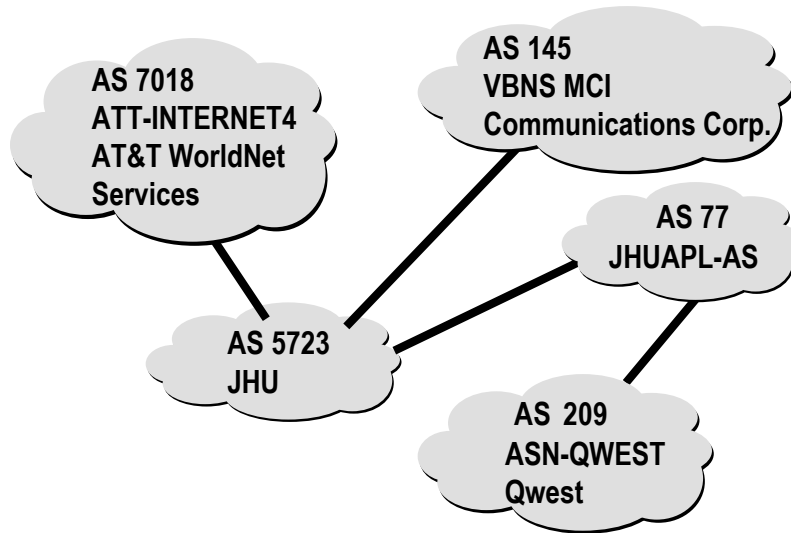
Inter-domain Routing

- **Inter-domain routing**
 - Based on a distributed system: composed of routers, grouped into administrative domains (autonomous systems —ASes)
 - Inter-domain routing protocols provide the means by which organizations exchange and propagate reachability information among themselves

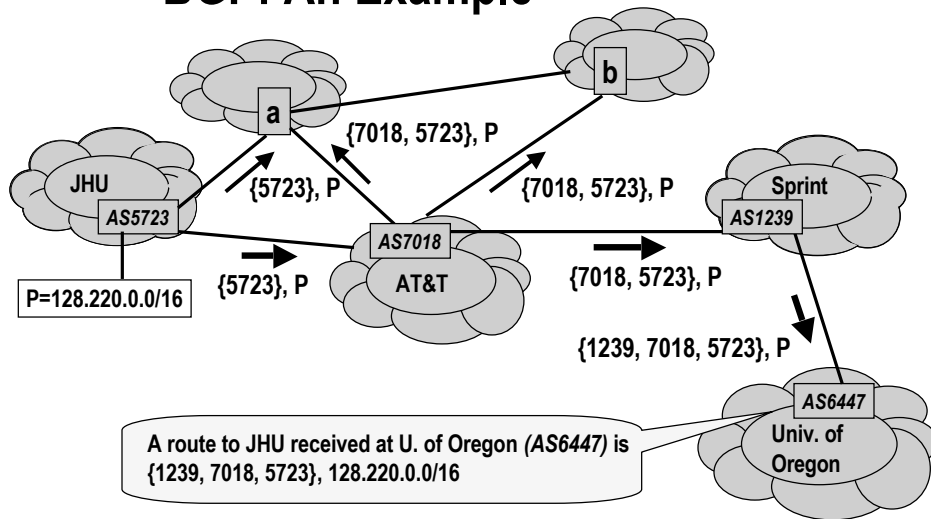
Border Gateway Protocol (BGP)

- **The *de facto* inter-domain routing protocol**
 - [Rekhter-et al, RFC1771, 1995]
- **Internet is considered as a collection of *Autonomous Systems*, each of which represents a portion of the network under single administrative control**
- **Border routers of different ASes connect directly with each other and establish BGP sessions**
- **Routing information is exchanged and propagated between ASes through routes**

Autonomous Systems: An Example



BGP: An Example



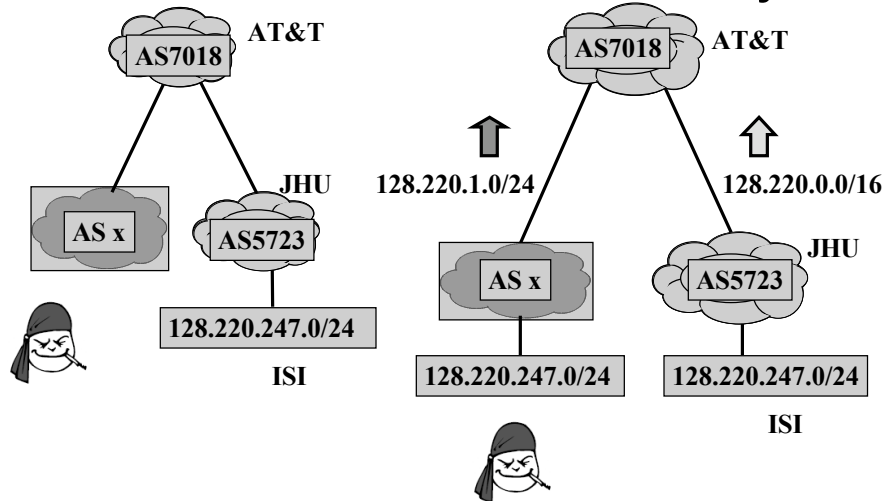
Security Concerns of BGP

- **Lack of secure means to verify the legitimacy and accuracy of its control traffic**
- **Vulnerable to malicious attacks and accidental misconfigurations**
- **Result in large segments of the internet becoming inaccessible**

Security Concerns in BGP (Cont'd)

- **Configuration errors:**
 - **Accidental insertion of routes into the global routing tables (origin misconfiguration)**
 - **Accidental propagation of routes that should be filtered out (export misconfiguration)**
- **Major BGP vulnerabilities**
 - **Messages are not guaranteed secrecy, integrity and freshness**
 - **Lack of origin authentication**
 - **Lack of path authentication**

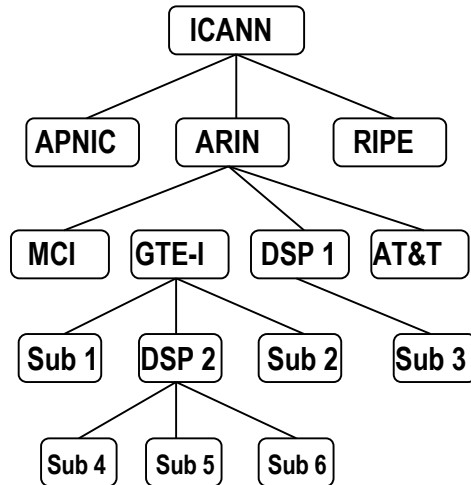
BGP Vulnerabilities: A Case Study



Secure BGP (SBGP)

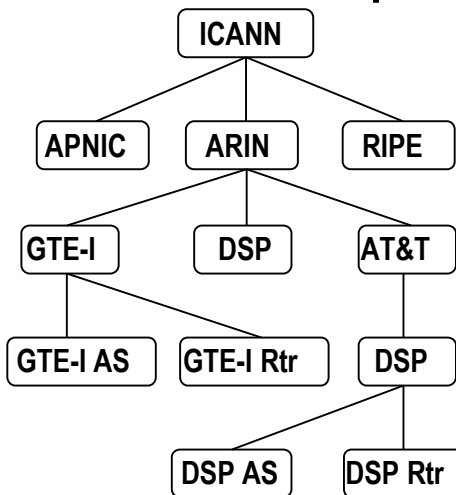
- [Kent-*et al*, IJSAC, 2000]
- Security for BGP is defined as correct operations of BGP speakers
- Three primary security mechanisms
 - Public key infrastructure (PKI)
 - Attestation
 - IPsec — to protect TCP connections

Address Space PKI



- Root (ICANN) -> registries -> ISPs -> DSPs or subscribers
 - Assignment of address space to organizations
 - The 1st type of certificate binds a public key to an organization and to a set of IP address prefixes

AS Ownership and Router ID PKI



- Root (ICANN) -> registries -> ISPs, DSPs or subscribers -> ASes and routers
 - The 2nd type of certificate binds a public key to an organization and a set of AS numbers
 - The 3rd type of certificate binds a public key to an AS number and to a BGP router ID

Attestations

- **Digitally signed statements**
 - Address attestation (AA)
 - Route attestation (RA)
- **Can be verified by participating entities**
- **Provide authenticity of the source of route announcement**
- **Protect the announcements themselves**

Discussions

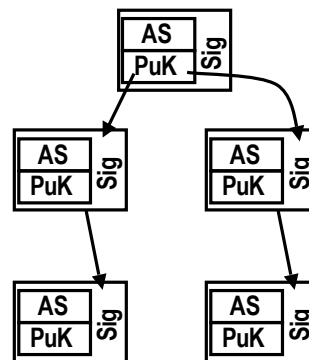
- **S-BGP**
 - Provide comparatively strong protection (origin and path)
 - Doesn't address the problem whether BGP rules and ASes' routing policies are correctly applied
- **Requirement for deployment**
 - Adoption of this technology by ISPs
 - PKI support by the registries that allocate AS numbers to ISPs and DSPs, and address prefixes to customers
 - High memory requirement and CPU utilization on routers

Secure Origin BGP

- [Ng, internet draft, 2002]
- A new message type: SECURITY
 - Carry security information within BGP protocol
 - Used to transport three types of certificates and a request format for requesting security certificates
- Three types of certificates
 - Entity certificate
 - Policy certificate
 - Authorization certificate

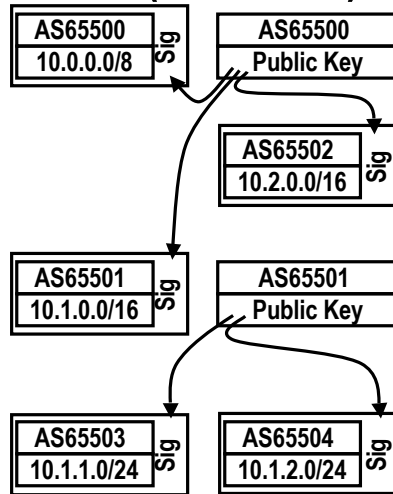
Entity Certificate (EntityCert)

- Each participant (entity) in the internet creates a public/private key pair
- Each participant then has these keys signed by a trusted third party
- The key and AS can be validated using the signer's public key
- This signed certificate is called an EntityCert



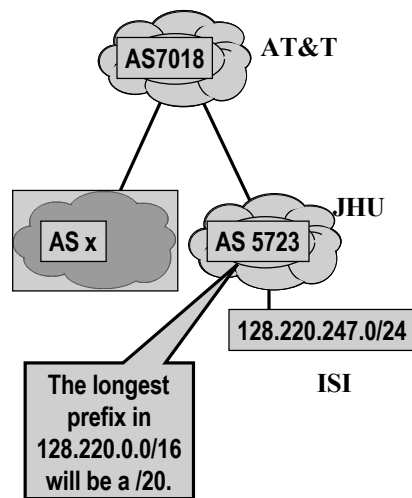
Authorization Certificate (AuthCert)

- If an AS authorizes another AS to advertise a given block of prefixes, it issues a separate certificate, signed using its private key, to indicate this authorization
- This is called an AuthCert



Policy Certificate (PolicyCert)

- Each AS also builds a certificate which contains policy information, and signs it with its private key
- This is called the PolicyCert
- The public key of AS, learned from the EntityCerts, can be used to validate each PolicyCert

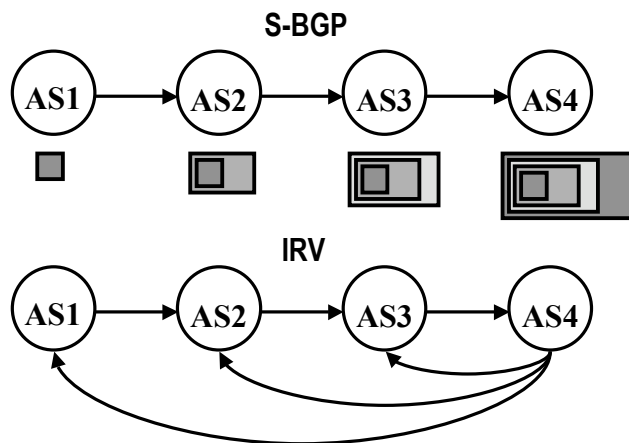


Discussions

- Entail less changes to current BGP systems
- Path authentication is not as strong as S-BGP
- Doesn't protect
 - The BGP transport connection
 - BGP attribute validity
 - Full validity of the AS_PATH — does not verify that the AS_PATH of any given route has not been modified in transit

Inter-domain Routing Validation (IRV)

- [Goodell-et al, NDSS, 2003]



IRV (Cont'd)

- **Each participating AS designates an IRV server for answering queries from other ASes**
- **Receiver-driven**
 - **Users query the IRV to validate received BGP data or to acquire additional route-relevant information**
- **IPsec or TLS can be used to ensure the integrity, authenticity, and timeliness of the queries and responses**