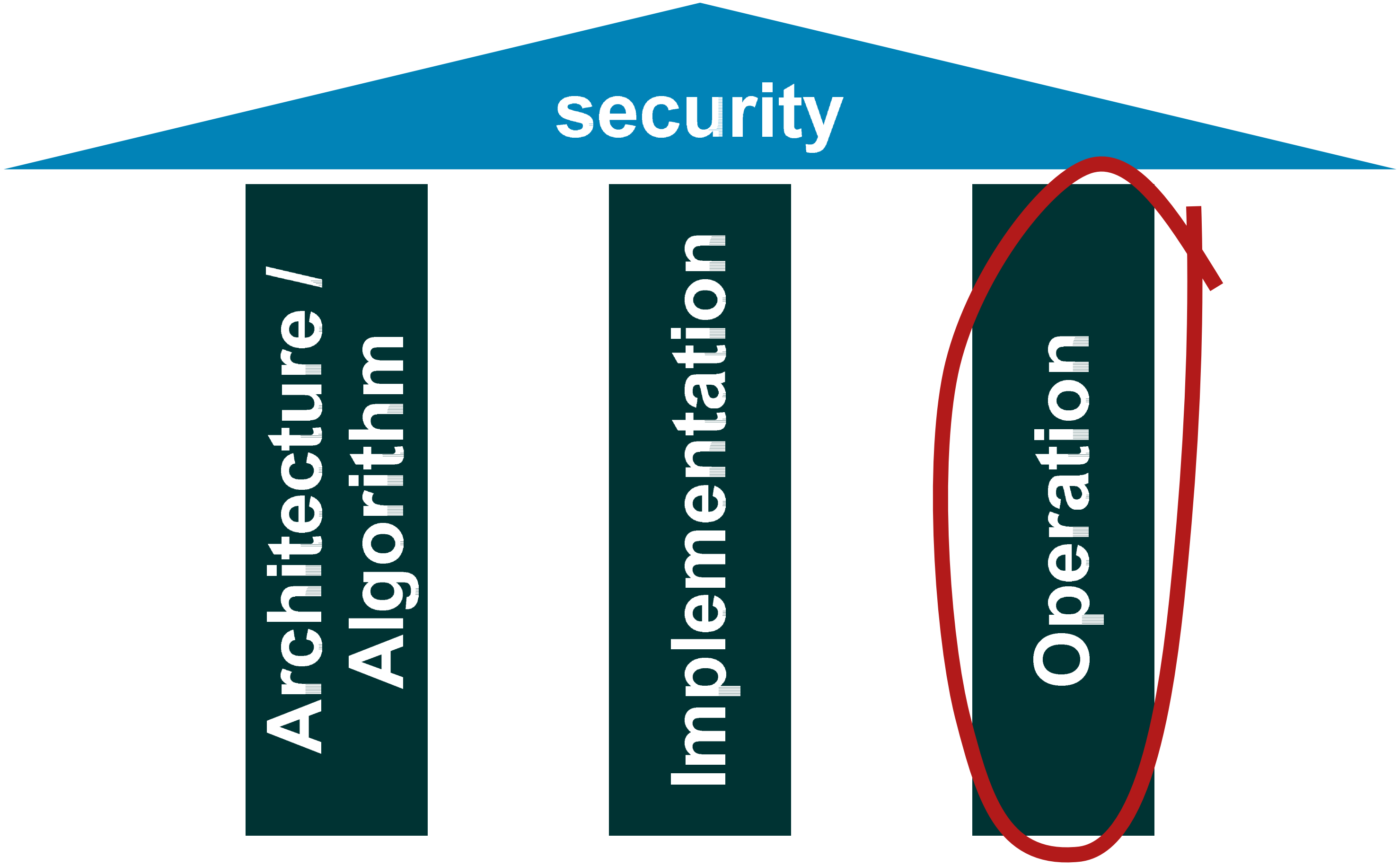
Detecting Router Abuse

Michael Behringer <mbehring@cisco.com>

Security Relies on Three Pillars



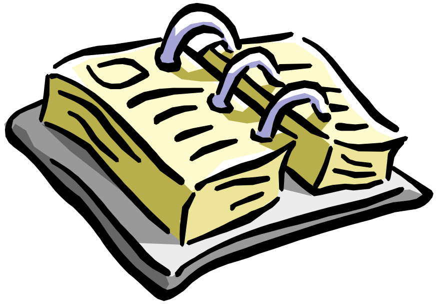
Break one, and all security is gone!

Goal: Detect Misconfigurations — *before* They Cause Problems

"There's the problem - you've got it set on 'BROIL.'"

Detecting Router Abuse

Agenda



- Threat model
- Overview: What we have, and what not
- Detection methods
  - Device based
  - Network based
- Preventing router abuse: Assorted ideas

The Traditional SP Threat Model

- Untrusted:
  - The SPs peers
  - The SPs upstreams
  - The SPs customers
- Trusted:
  - The SPs operation

The SP Threat Model Used Here

- Untrusted:

  The SPs peers

  The SPs upstreams

  The SPs customers

  The SPs operation

- Trusted:

  ~~The SPs operation~~

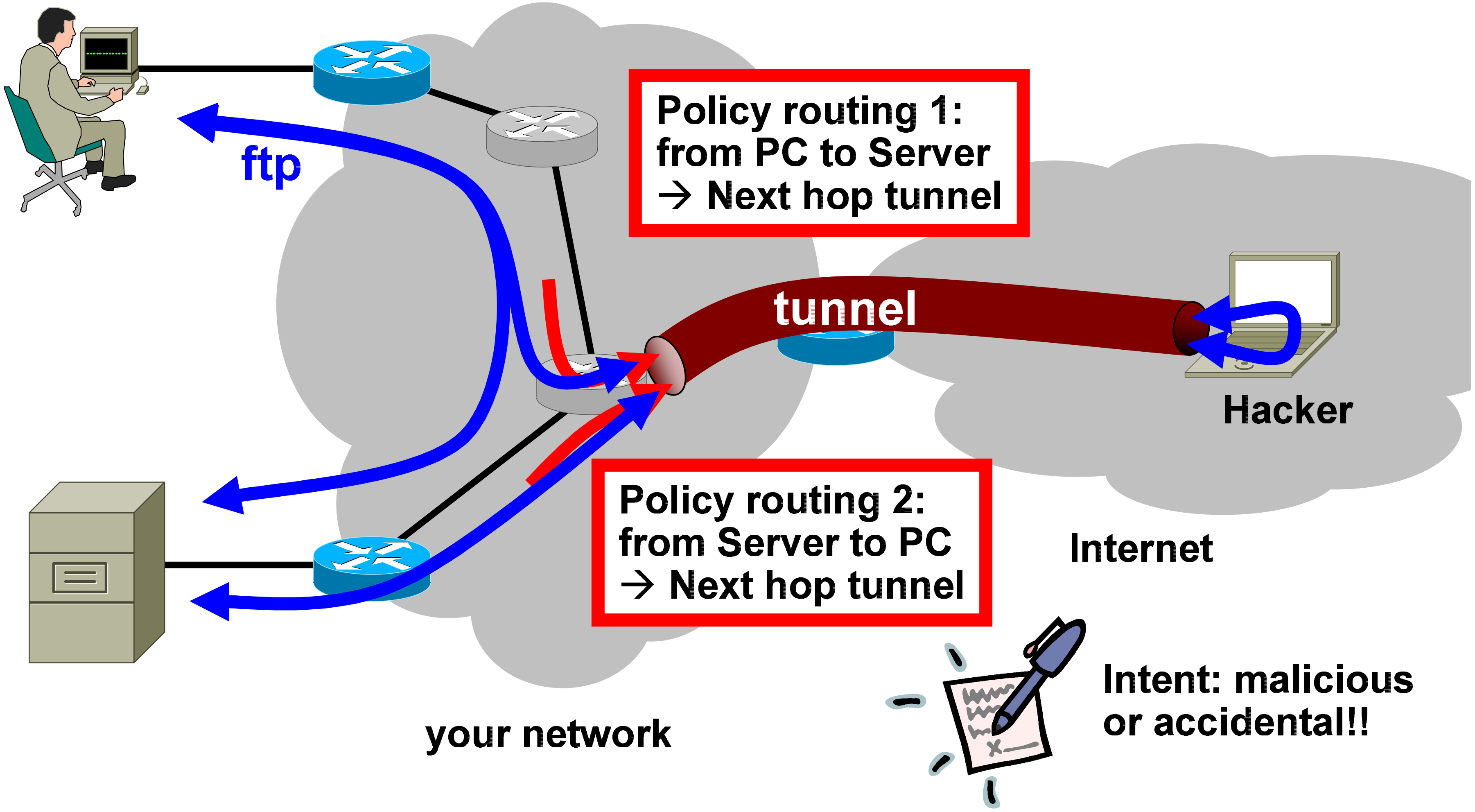
  …

  well, who can we trust???

Focus here:

• Insider attacks, both malicious and accidental (main focus);
• Outsider attacks (hacked router)

The Threat Model, Part 1: Unauthorised Configuration on a Router

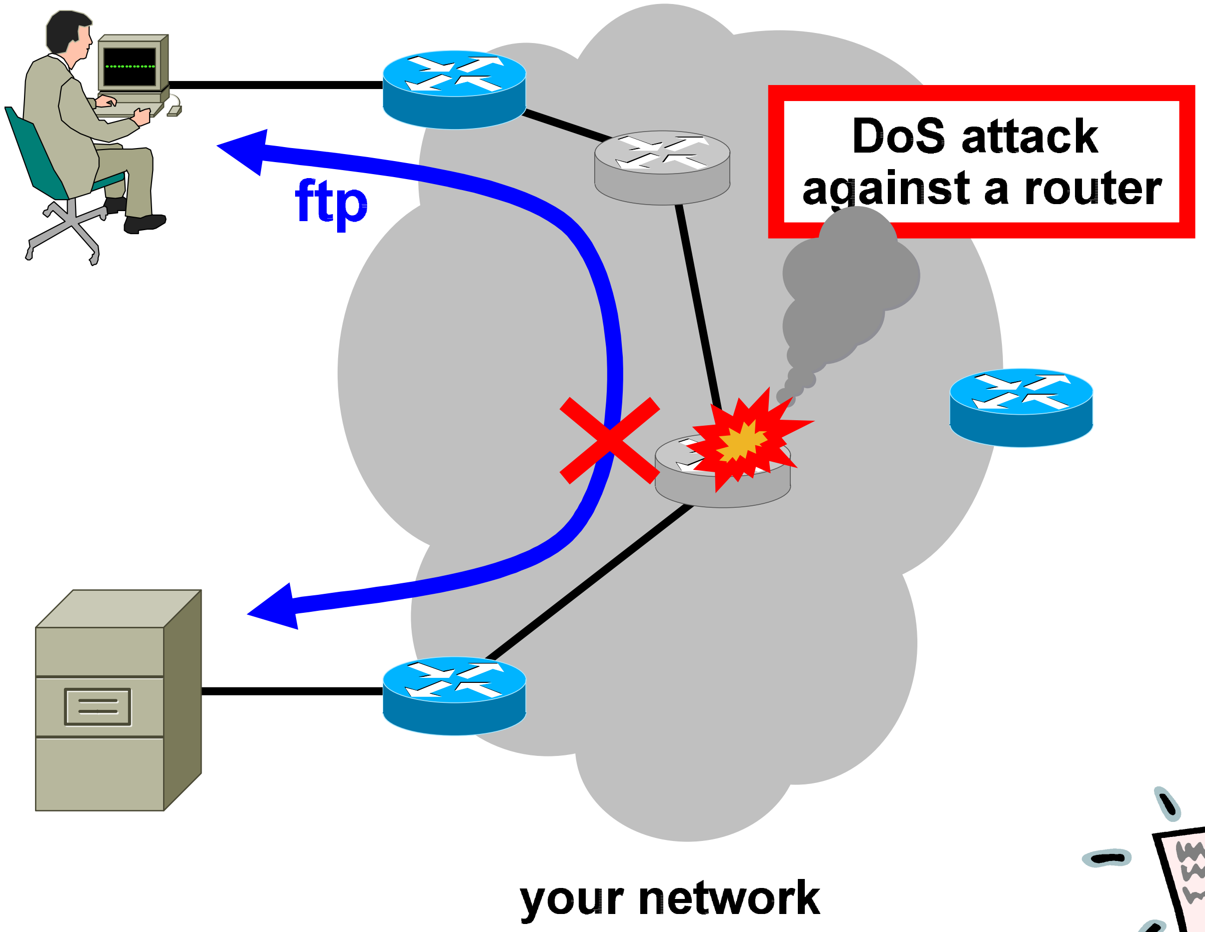


Policy routing 1:
from PC to Server
→ Next hop tunnel

Policy routing 2:
from Server to PC
→ Next hop tunnel

Intent: malicious or accidental!!

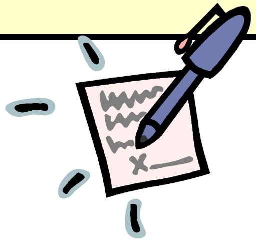
The Threat Model, Part 2: DoS



Intent: malicious or accidental!!

The Threat Model, Summary

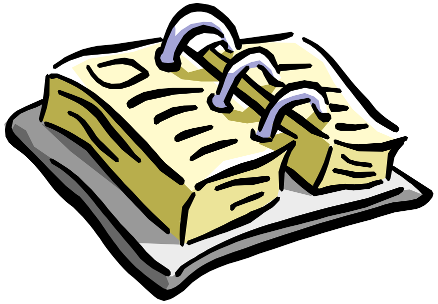
	Malicious	Accidental
Configuration	Configuration modifications	misconfiguration
DoS	packet floods, protocol attacks, etc.	misconfiguration, routing errors, etc.



both internal (trusted) and
external (untrusted) attack sources

Detecting Router Abuse

Agenda



- Threat model
- Overview: What we have, and what not
- Detection methods
  - Device based
  - Network based
- Preventing router abuse: Assorted ideas

What You Already Know And Have Implemented ;-)

- Disable unused services (http, finger, …)
- Use AAA and strong passwords
- Use application ACLs for SNMP, telnet, etc
- Use interface ACLs (infrastructure ACLs)
- MD5 and key chains for routing
- BGP GTSM (TTL security mechanism)
- Use secure protocols (SSH, SCP, SNMPv3, …)
- Route filters: bogons, private, unallocated, your own
- Traffic filters: your own, special cases, etc.
- Secure your services (AAA, DNS, NTP, FTP, …)
- Physical security (no access to console)
- no service password-recovery
- …

Caveat:
Everybody with enable access can circumvent / misconfigure all of those!!

The Old Model to Secure Infrastructure

- Secure each router

  SSH, AAA, access lists, routing authentication, etc…

- Missing:

  Detection of intrusions

  Detection of misconfigurations

  Detection of incorrect operation

- In other words: Assume …

  … you secured the router correctly

  … your router has no bugs

  implementation

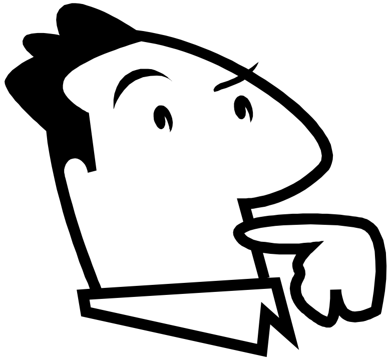
  … unauthorised people can't get in

  … authorised people make no mistakes

  … authorised people have no malicious intent

  operation

Shortcomings of the Old Model



- Reliance on
  - correct router configuration
  - router being bug free
- No / limited configuration control
  - assume malicious access not possible
  - assume authorised people make no mistakes
  - … and they have no malicious intentions
- Often no / insufficient device monitoring
  - login attempts?
  - config changes?

Where is the master config?

- On your router?
- On the NMS system?
- On some UNIX box? (long live perl & expect)
- On the GUI?
- In your head? In someone else's head?
- Everywhere?
- Nowhere?

So, in case of doubt, where do you check?

How do I change the config?

- CLI (where again was the master config?)
- NMS
- in band / out of band
- Through a central server
  (“nobody touches the box directly”)
- Concept of “least privilege”:

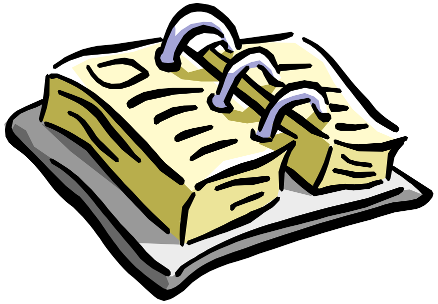
  Operator gets only the access rights he strictly needs
- Secondary question: Which protocol to use?

  SSH, and SCP. Of course! ☺

  copy scp: flash:

Detecting Router Abuse

Agenda



- Threat model
- Overview: What we have, and what not
- Detection methods
  - Device based
  - Network based
- Preventing router abuse: Assorted ideas

What to Control:

- Configuration changes
- Status changes (interface down)
- Hardware changes

  E.g., Flash Cards, USB token!
- OS changes

  Or, parts of the OS:

  - IPS signature description files (.sdf)

  - Flexible packet matching: Packet descriptors

  - …

General Principle for Logging: “Dual Control”

- Operations team:

  Router management, troubleshooting, configuration, …

  - no (write) access to security logs!!!

- Security team:

  Control of logging system

  Control of AAA, user authentication and authorisation

  Log evaluation

“Don’t let the cat guard the sausage”

Device Based Detection: Logging, logging, logging

- Log everything
- Look at the logs (!)
  - Need automation. Yes!!
  - Need automated alarms. For EVERY alert. (well…)
  - Automation: Never ending task
- Who made which change when?
  - And maybe, why?

“Knowing the murderer is not a solution for the victim;
but it does help the community to survive.”

Device Based: Configuration Verification

- Has your config changed?

  Trigger: `%SYS-5-CONFIG_I: Configured from console by x`

  Not perfect: Syslog messages can be filtered, routed to null0, …

  → Define periodic downloads, compare to master config

- If so, was it authorised?

  If available, link to provisioning system

  NOC needs to manually acknowledge the alarm

  This is work!

Tools for Configuration Comparison

- Rancid

  http://www.shrubbery.net/rancid/

  http://www.nanog.org/mtg-0310/rancid.html

- Tool

  http://tool.sourceforge.net

  http://www.nanog.org/mtg-0310/pdf/mcphersonpanel.pdf

- RAT

  http://www.cisecurity.org/

  http://www.nanog.org/mtg-0310/pdf/mcphersonpanel.pdf

- ISC Tools

  ftp://ftp.isc.org/isc/toolmakers/

  http://www.nanog.org/mtg-0210/abley.html

Device Based: SNMP Polling

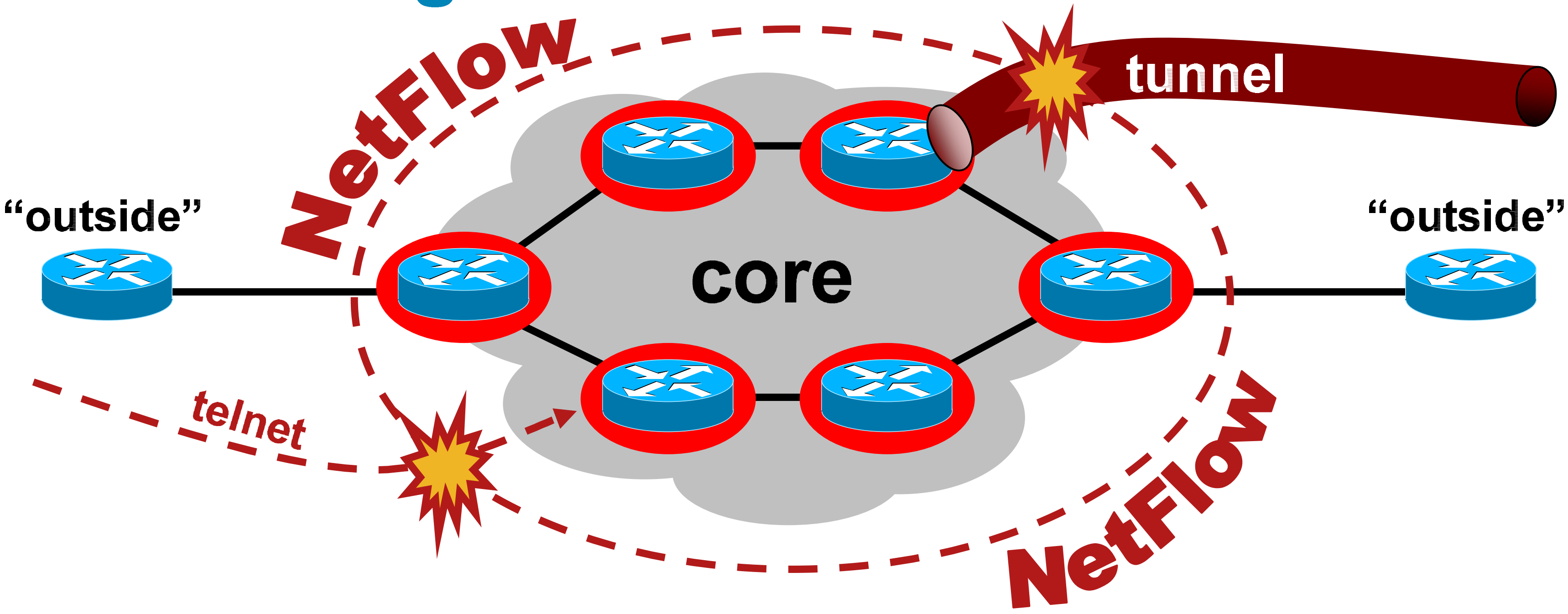
- Poll configuration related variables
- Compare with known "good" values
- Or, alert on change
- Example:

  Number of interfaces (IfTable): To detect GRE tunnels

Device Based: Command Authorisation

- Who did what when?

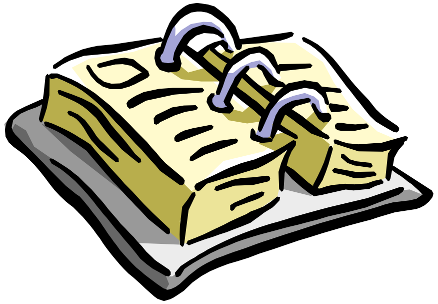
Network Based: Monitoring Flows into the Network



- Monitor flows where <src> or <dst> = <your core>
- Define "okay flows": BGP (to some routers), ICMP, …
- Most others indicate a security issue → Alert

Detecting Router Abuse

Agenda



- Threat model
- Overview: What we have, and what not
- Detection methods
  - Device based
  - Network based
- Preventing router abuse: Assorted ideas

“Prevention” is a Big Word!

“Who is authorised to configure,
is also authorised to misconfigure.”

- You cannot *prevent* misconfigurations
- Main focus: Detection
- However, you can make misconfiguration *harder*
  - Automation
  - Consistency checks

First: Physical Security

- Every router / switch: Physically secured

    Access control to room, monitoring

- Concerns:

    Password recovery → Config changes

    Theft

    Sniffing

No service password-recovery

- Different implementations
  - -When password recovery → erase NVRAM
  - -Password recovery impossible (really!)
- Where available: Use It!
- This makes it hard to intrude into a router, even with physical access!

AAA Authorisation

- Know who did what when
- Reactive, but a deterrent for malicious changes

CLI Views

- Role based device management

Summary

- Still essential: Securing every router separately

  SSH, AAA, access lists, SNMPv3, disable services, etc.

- New Infrastructure paradigms:

  Make routers inaccessible (Infrastructure ACLs)

  Isolation of the IGP

- Secure Operations

  Log changes, login attempts, track snmp variables, etc.

  Define configuration management

Outlook on Router Security

- Management plane separated from data plane

  → Users have no access to management plane

- Control plane separated from data plane

  → Users cannot affect routing, ntp, etc.

- Configuration management

  Problem: Operator has right to configure → May also misconfigure

  Must be enforced by operational procedures, dual control

  This is hard!

Q and A