Outline

Even web more risks

Firewalls and NAT boxes

Announcements intermission

Intrusion detection systems

Misconfiguration problems (A5)

- Default accounts
- Unneeded features
- Framework behaviors
  - Don't automatically create variables from query fields

Openness tradeoffs

- Error reporting
  - Few benign users want to see a stack backtrace
- Directory listings
  - Hallmark of the old days
- Readable source code of scripts
  - Doesn't have your DB password in it, does it?

Using vulnerable components (A9)

- Large web apps can use a lot of third-party code
- Convenient for attackers too
  - OWASP: two popular vulnerable components downloaded 22m times
- Hiding doesn't work if it's popular
- Stay up to date on security announcements

Clickjacking

- Fool users about what they're clicking on
  - Circumvent security confirmations
  - Fabricate ad interest
- Example techniques:
  - Frame embedding
  - Transparency
  - Spoof cursor
  - Temporal "bait and switch"
Crawling and scraping
- A lot of web content is free-of-charge, but proprietary
  - Yours in a certain context, if you view ads, etc.
- Sites don’t want it downloaded automatically (web crawling)
- Or parsed and used for another purpose (screen scraping)
- High-rate or honest access detectable

Outline
- Even web more risks
- Firewalls and NAT boxes
- Announcements intermission
- Intrusion detection systems

Internet addition: middleboxes
- Original design: middle of net is only routers
  - End-to-end principle
- Modern reality: more functionality in the network
  - Security is one major driver

Security/connectivity tradeoff
- A lot of security risk comes from a network connection
  - Attacker could be anywhere in the world
- Reducing connectivity makes security easier
- Connectivity demand comes from end users

What a firewall is
- Basically, a router that chooses not to forward some traffic
  - Based on an a-priori policy
- More complex architectures have multiple layers
  - DMZ area between outer and inner layers, for outward-facing services

Inbound and outbound control
- Most obvious firewall use: prevent attacks from the outside
- Often also some control of insiders
  - Block malware-infected hosts
  - Employees wasting time on Facebook
  - Selling sensitive info to competitors
  - Nation-state Internet management
- May want to log or rate-limit, not block
Default: deny
- Usual whitelist approach: first, block everything
- Then allow certain traffic
- Basic: filter packets based on headers
- More sophisticated: proxy traffic at a higher level

IPv4 address scarcity
- Design limit of $2^{32}$ hosts
  - Actually less for many reasons
- Addresses becoming gradually more scarce over a many-year scale
- Some high-profile exhaustions in 2011
- IPv6 adoption still very low, occasional signs of progress

Network address translation (NAT)
- Middlebox that rewrites addresses in packets
- Main use: allow inside network to use non-unique IP addresses
  - RFC 1918: 10.*, 192.168.*, etc.
  - While sharing one outside IP address
- Inside hosts not addressable from outside
  - De-facto firewall

Packet filtering rules
- Match based on:
  - Source IP address
  - Source port
  - Destination IP address
  - Destination port
  - Packet flags: TCP vs. UDP, TCP ACK, etc.
- Action, e.g. allow or block
- Obviously limited in specificity

Client and server ports
- TCP servers listen on well-known port numbers
  - Often < 1024, e.g. 22 for SSH or 80 for HTTP
- Clients use a kernel-assigned random high port
- Plain packet filter would need to allow all high-port incoming traffic

Stateful filtering
- In general: firewall rules depend on previously-seen traffic
- Key instance: allow replies to an outbound connection
- See: port 23746 to port 80
- Allow incoming port 23746
  - To same inside host
- Needed to make a NAT practical
Circuit-level proxying
- Firewall forwards TCP connections for inside client
- Standard protocol: SOCKS
  - Supported by most web browsers
  - Wrapper approaches for non-aware apps
- Not much more powerful than packet-level filtering

Application-level proxying
- Knows about higher-level semantics
- Long history for, e.g., email, now HTTP most important
- More knowledge allows better filtering decisions
  - But, more effort to set up
- Newer: “transparent proxy”
  - Pretty much a man-in-the-middle

Tunneling
- Any data can be transmitted on any channel, if both sides agree
- E.g., encapsulate IP packets over SSH connection
  - Compare covert channels, steganography
- Powerful way to subvert firewall
  - Some legitimate uses

Outline
- Even web more risks
- Firewalls and NAT boxes
- Announcements intermission
- Intrusion detection systems

Note: more readings this week
- More details on how to set up firewalls
- Burglar alarms and “mimicry” attack on IDSes
- Containing high-speed worms
- Virus evolution in 2012

Research project status
- Next (and final) meetings will be 11/30-12/4
- Next (and final) progress reports 12/4
- Presentations 12/7-12/16
Exercise set 4

- Posted this morning
- Covers random numbers, middleboxes, and networks
- Due next Tuesday 11/24 11:55pm

Hands-on Assignment 2

- All running as of Saturday
  - Email Nishad if you don't have a VM
- Choice of extended due date:
  - Monday 11/23
  - Friday 11/27
  - Sunday 11/29
  - Monday 11/30

Outline

- Even web more risks
- Firewalls and NAT boxes
- Announcements intermission
- Intrusion detection systems

Basic idea: detect attacks

- The worst attacks are the ones you don't even know about
- Best case: stop before damage occurs
  - Marketed as “prevention”
- Still good: prompt response
- Challenge: what is an attack?

Network and host-based IDSes

- Network IDS: watch packets similar to firewall
  - But don't know what's bad until you see it
  - More often implemented offline
- Host-based IDS: look for compromised process or user from within machine

Signature matching

- Signature is a pattern that matches known bad behavior
- Typically human-curated to ensure specificity
- See also: anti-virus scanners
Anomaly detection
- Learn pattern of normal behavior
- "Not normal" is a sign of a potential attack
- Has possibility of finding novel attacks
- Performance depends on normal behavior too

Recall: FPs and FNs
- False positive: detector goes off without real attack
- False negative: attack happens without detection
- Any detector design is a tradeoff between these (ROC curve)

Signature and anomaly weaknesses
- Signatures
  - Won’t exist for novel attacks
  - Often easy to attack around
- Anomaly detection
  - Hard to avoid false positives
  - Adversary can train over time

Base rate problems
- If the true incidence is small (low base rate), most positives will be false
  - Example: screening test for rare disease
- Easy for false positives to overwhelm admins
  - E.g., 100 attacks out of 10 million packets, 0.01% FP rate
    - How many false alarms?

Adversarial challenges
- FP/FN statistics based on a fixed set of attacks
- But attackers won’t keep using techniques that are detected
- Instead, will look for:
  - Existing attacks that are not detected
  - Minimal changes to attacks
  - Truly novel attacks

Wagner and Soto mimicry attack
- Host-based IDS based on sequence of syscalls
  - Compute $\mathcal{A} \cap \mathcal{M}$, where:
    - $\mathcal{A}$ models allowed sequences
    - $\mathcal{M}$ models sequences achieving attacker’s goals
  - Further techniques required:
    - Many syscalls made into NOPs
    - Replacement subsequences with similar effect
Next time

Malware and network denial of service