Network Defense
CMSC 414

November 27, 2017
Defense is Hard

We’ve seen a lot of attacks, across:

- Physical Layer
- Link Layer
- Network Layer
- Transport Layer
- Application Layer

An attacker needs one way to accomplish goal:

- Denial-of-Service
- Information Extraction
- Malware Propagation
- Persistence

A defender must protect *all layers* against *all possible attacks* ⇒ Defense-in-Depth
Keeping Software Up-to-Date

Bugs constantly found in software

(Often the same sorts of bugs we’ve seen many times before)

Some are security vulnerabilities

May take a while, but most end up being patched

Best practice: Apply patches/updates when they come out!

However...
Why Admins (and Users) Don’t Patch Systems

Applying patches takes time and effort

Often, this means downtime

Most organizations have third-party (or in-house) software
  ⇒ May break with upgrade

Testing takes more time/effort/equipment
  ⇒ Can’t (or at least shouldn’t) test in production!

Sometimes policy prevents updating

Sometimes patches/updates can’t get to you
Configuration Management

Even good software can be used badly

What interfaces does it listen on? What ports?

Is security enabled? Are the defaults actually secure?

Is automatic updating enabled?

Are there default users and default passwords?

Do we actually need to run everything we’re running?

Who can install software?

Is your configuration consistent across machines?
Centralized Configuration

Technology to the rescue!

*Preboot Execution Environment (PXE)*
- PXE-enabled NIC
  - Host configures itself on boot from network resources

*Puppet*
- Ruby
  - Supports push changes

*Chef*
- Ruby
  - Data can be accumulated/propagated at runtime

*Ansible*
- Less language dependence
- Hosts only require ssh and python
- Tries to make operations *idempotent*
Operational Security

*Practices* that support security

User training

How do you avoid being phished?

How do you store/dispose of sensitive materials?

Are you exposing sensitive information in what you say/do/post?

Are you installing malicious software or carrying malicious devices?

Frequently *mandatory training*
  ⇒ Frequently forgotten as soon as training complete
Filtering

Can’t rely on users behaving correctly

Can’t rely on hosts being configured securely

Prevent *bad stuff* from *getting in*

Prevent *secret stuff* from *getting out*

⇒ **Filter** traffic
Simple Firewalls

Inside
10.0.0.0/8

Outside
*/*

daddr=10.255.255.255

ICMP Echo Request

dport=1337

saddr=10.0.1.3

external pkt w/ internal saddr

saddr=128.1.56.2

daddr=20.3.4.17

attempted connex to non-std port

external pings

Simple filtering on IP addresses/ports/protocols
Some knowledge of connection state
Spam Filters

Email goes to a well-known port

Should also only go to specific servers

Can filter these at firewall
  ▶ Wrong daddr/dport? ⇒ drop it
  ▶ Correct daddr/dport ⇒ route through spam filter

Spam filter can check for
  ▶ Known malware signatures
  ▶ Suspicious source domains
  ▶ Unusual commands
  ▶ Rapid multiple deliveries

Some spam filters can *defang* attachments
Deep Packet Inspection

Spam filtering is a type of **Deep Packet Inspection** (DPI)

Can operate at any layer from 3 to 7

Have to reconstruct packets (if fragmented)

May have to reconstruct TCP sessions

Application-layer inspection may need separate node, like spam filter

*DPI is very expensive*

- In-line only in limited circumstances
- Can be used in parallel to normal packet delivery (Audit)
- Often develops filtering rules for firewall
Censorship and Wiretaps

Sometimes government regulations require *censorship* of traffic

- Unlawful content
- Forbidden Internet destinations

May be required by edge networks for liability reasons

Some organizations do content filtering on their own
  ⇒ Not technically censorship, but similar tools/effects

Most countries also require provisions for *Lawful Intercept* (LI)

- Install/enable wiretaps with a warrant
- Capability built into most commercial routers/firewalls
- Some jurisdictions might require this to be always-on

Note that both of these can be mis-used
Intrusion Detection

Usual goal: *Keep bad stuff/people out*

Part of defense-in-depth is recognizing this is not always possible

**Intrusion Detection System (IDS)** looks for
- Questionable traffic egressing (DPI)
- New users (change detection on LDAP/AD)
- New services listening (nmap)
- New software installed (change detection on all hosts)
- Errors/abnormalities in log files

May or may not have a *model* for the intruder
- *Misuse detection* checks for known adversarial behavior
- *Anomaly detection* checks for anything abnormal
Challenges

Not all intrusions produce signatures (eg, hidden microphones)

Traffic is noisy — hard to distinguish attacks & garbage

*False positives*

- Low FP rate still produces more FP than TP
- Investigating each positive result takes time/effort
- Start tuning out positives (eg TSA screenings, polygraphs)

*False negatives*

- False sense of security
- FN rate may never be known
- Eventually (hopefully) discover intrusions through other means
Challenges

Constantly changing signatures
- New software ⇒ new attacks
- Same software ⇒ new attacks
- Can’t get rid of old software ⇒ have to keep old signatures

IDS needs updating, like any software
⇒ All update issues still apply

A lot of traffic is *opaque*
- Encryption
- Binary formats

Makes inspection much more difficult, if not impossible
Honeypots/Honeynets

What are attackers up to?

Why not *ask them*?

Set up servers that *look real*

- Easy (but not *too* easy) to break into
- Nothing important on them
- Restricted ability to send traffic
- Called a **Honeypot**

Observe what the attacker does

Build model of attacker for IDS

Can set up an entire shadow network for attacker to explore

⇒ **Honeynet**
Bring-Your-Own-Device

Many organizations encourage Bring-Your-Own-Device (BYOD)

Some require it

Mobile phones, laptops, tablets, . . .

Reduces equipment expenses

Allows/encourages employees to work 24/7

But
  ▶ Personal devices harder to secure
  ▶ Can’t enforce patch/upgrade schedule
  ▶ Harder to track what devices are permitted on network

People who must secure the network have no control over BYOD
Layered Networks

Not all hosts equal:

- Some servers must be externally visible
- Some servers only for internal use
- Most desktops should only be internally accessible
- BYOD should not be trusted
- Some hosts will have sensitive information

Ideally, we can separate hosts with different needs

Innermost network has most sensitive/vulnerable stuff

Each network has its own defenses

Further in you go

- Less traffic in/out
- Fewer machines
- Ability to do *more thorough examination*
Network Architecture

- Internet
- Public Services
- Internal Services
- Trusted Desktops
- Sensitive Resources
- BYOD
- IDS
- Guests
- Highly Sensitive Resources
- Airgap