Intrusion Detection

CS 161/194-1
Anthony D. Joseph
September 14, 2005

Outline

- History
- Network-based Host Compromise
- Host-based Network Intrusion Detection
  - Signature-based
  - Anomaly-based
- Distributed Network Intrusion Detection
  - Honeypots
  - Tarpits
- An attack against an IDS

Intrusion Detection History

- Detecting attempts to penetrate our systems
  - Used for post-mortem activities
  - Related problem of extrusion (info leaking out)
- In pre-network days (centralized mainframes)…
  - Primary concern is abuse and insider information access/theft
  - Reliance on logging and audit trails
- But, highly labor intensive to analyze logs
  - What is abnormal activity?
  - Ex: IRS employees snooping records
  - Ex: Moonlighting police officers

Network-based Host Compromises

- How do remote intruders gain access?
- They attempt network-based attacks that exploit OS & app bugs
  - Ex: Denial of service, spyware install, zombie, …

Host-based Network Intrusion Detection

- At each host, monitor all incoming and outgoing network traffic – for each packet:
  - Analyze 4-tuple and protocol
  - Examine contents
  - …
- Challenge: Separate "signal" from "noise"
  - *Signal* is an attack (intrusion)
  - *Noise* is normal "background" traffic
  - Assumption: can separate signal and noise…

Some Challenges

- What is normal traffic?
  - Server, desktop, PDA, PDA/phone, …
  - My normal traffic ? your normal traffic
  - Lots of data for servers
- Why do we need sufficient signal and noise separation?
  - To avoid too many false alarms!
- What happens if signals are missed?
  - Possible intrusion!

Some Common False Positives

- Proximity probes
  - Website load balancers will probe your machine for proximity
  - Connect to website hosted by mirror-image.com, and >10 load balancers in 6 countries probe your machine
- Stale IP caches
  - Using dynamic IP addresses, you may get the “old” address of someone who was running a P2P app
  - Peers continue to try to “re-connect”
- Web posts with dynamic IP addresses
  - Spiders crawl machine currently using IP address

Lots and Lots of Data!!

- Network trace from Win2K desktop

```
ZoneAlarm Logging Client v3.7.202
Windows 2000-5.0.2195-Service Pack 4-SP
type,date,time,source,destination,transport
FWIN,2004/01/15,13:17:38 -8:00 GMT,216.183.33.67:42645,128.32.168.229:6129,TCP (flags:S)
FWOUT,2004/01/15,13:18:00 -8:00 GMT,128.32.168.229:5000,68.26.217.204:5000,UDP
FWIN,2004/01/15,13:42:38 -8:00 GMT,61.178.60.11:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWIN,2004/01/15,13:42:48 -8:00 GMT,62.177.227.10:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWIN,2004/01/15,13:48:12 -8:00 GMT,128.32.41.80:1040,128.32.168.229:38293,UDP
FWIN,2004/01/15,13:58:30 -8:00 GMT,24.224.253.230:2446,128.32.168.229:6129,TCP (flags:S)
FWIN,2004/01/15,14:04:40 -8:00 GMT,80.116.4.42:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWOUT,2004/01/15,14:04:44 -8:00 GMT,128.32.168.229:5000,68.26.217.204:5000,UDP
FWIN,2004/01/15,14:07:36 -8:00 GMT,210.217.129.194:3598,128.32.168.229:1433,TCP (flags:S)
FWIN,2004/01/15,14:15:00 -8:00 GMT,128.32.30.70:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWIN,2004/01/15,14:23:20 -8:00 GMT,80.56.148.243:0,128.32.168.229:0,ICMP (type:3/subtype:1)
FWIN,2004/01/15,14:41:48 -8:00 GMT,194.23.44.215:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWIN,2004/01/15,14:43:08 -8:00 GMT,61.64.246.192:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWOUT,2004/01/15,14:43:16 -8:00 GMT,128.32.168.229:5000,68.26.217.204:5000,UDP
FWIN,2004/01/15,15:02:00 -8:00 GMT,128.32.168.21:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWIN,2004/01/15,15:06:28 -8:00 GMT,81.185.244.166:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWIN,2004/01/15,15:43:46 -8:00 GMT,217.255.55.163:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWOUT,2004/01/15,15:44:16 -8:00 GMT,128.32.168.229:5000,68.26.217.204:5000,UDP
FWIN,2004/01/15,15:50:06 -8:00 GMT,65.78.10.110:3071,128.32.168.229:3410,TCP (flags:S)
FWIN,2004/01/15,15:59:42 -8:00 GMT,202.42.49.198:0,128.32.168.229:0,ICMP (type:8/subtype:0)
FWIN,2004/01/15,16:07:40 -8:00 GMT,68.22.89.249:4088,128.32.168.229:1433,TCP (flags:S)
FWIN,2004/01/15,16:08:36 -8:00 GMT,193.95.219.45:0,128.32.168.229:0,ICMP (type:3/subtype:1)
FWIN,2004/01/15,16:23:50 -8:00 GMT,67.37.40.15:4299,128.32.168.229:3410,TCP (flags:S)
FWOUT,2004/01/15,16:24:16 -8:00 GMT,128.32.168.229:5000,68.26.217.204:5000,UDP
```

Trace Analysis

- ZoneAlarm Logging Client v3.7.202
- Windows 2000-5.0.2195-Service Pack 4-SP
- type,date,time,source,destination,transport
- FWIN,2004/01/15,13:17:38 -8:00 GMT,216.183.33.67:42645,128.32.168.229:6129,TCP (flags:S)
- FWOUT,2004/01/15,13:18:00 -8:00 GMT,128.32.168.229:5000,68.26.217.204:5000,UDP
- FWIN,2004/01/15,13:42:38 -8:00 GMT,61.178.60.11:0,128.32.168.229:0,ICMP (type:8/subtype:0)
- FWIN,200 GMT,62.1
- FWIN,200 GMT,128.32.41.80:1040,128.32.168.229:38293,UDP

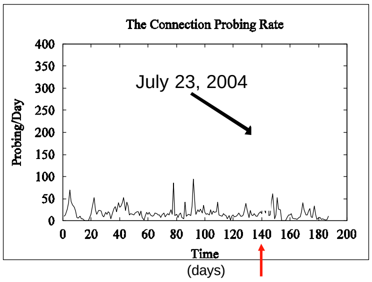
b2b-33-67.ip.granderiver.com

“ping” probe

Used by the Dameware remote admin sw (old versions have a bug allowing unauthorized login). Dameware also installed by some viruses

Analyzing Host-based Trace Data

- TCP connection probes on port 445
- Day 0 is 2003/03/04



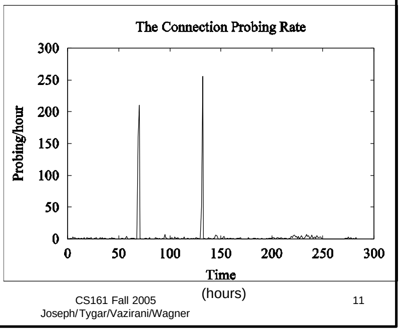
July 23, 2004

(days)

MSBlaster Worm

MSBlaster in Detail

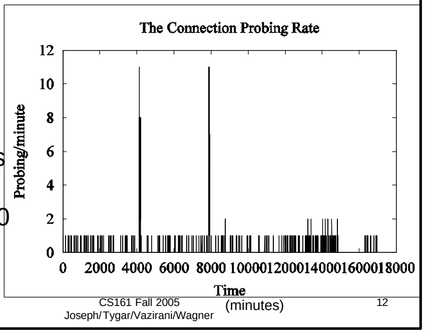
- TCP 445 probes/hr
- Hour 0 is 15:20 on 2003/07/20



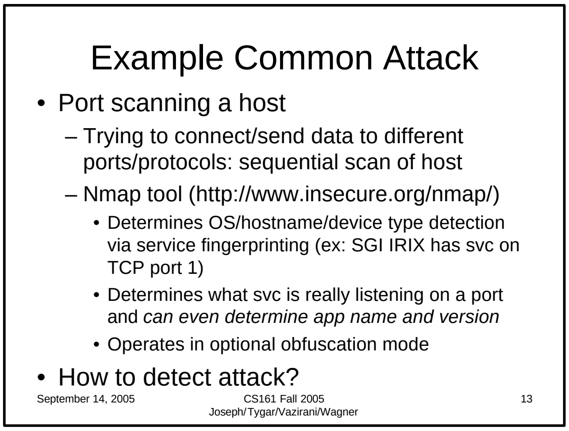
(hours)

MSBlaster in More Detail

- TCP 445 probes / 10 min
- Minute 0 is 15:20 on 2003/07/20

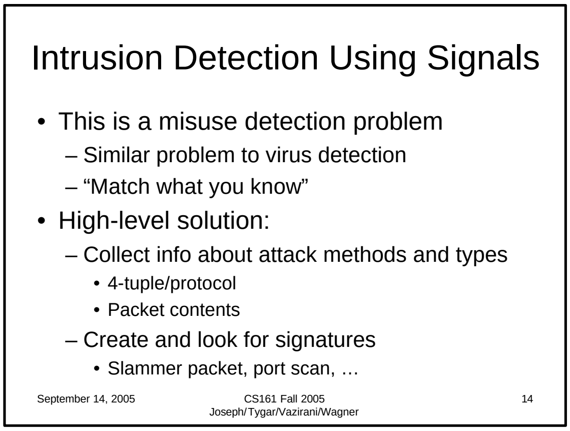


(minutes)



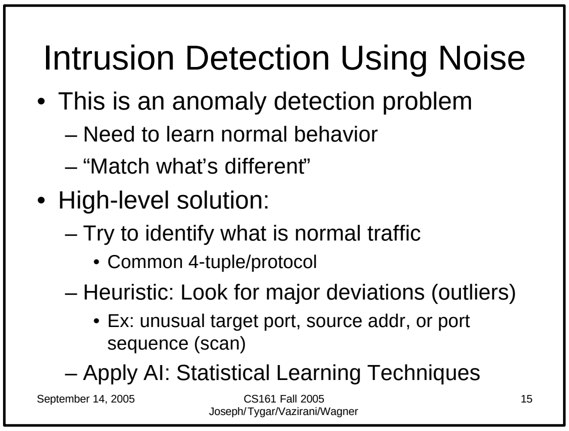
Example Common Attack

- Port scanning a host
  - Trying to connect/send data to different ports/protocols: sequential scan of host
  - Nmap tool (http://www.insecure.org/nmap/)
    - Determines OS/hostname/device type detection via service fingerprinting (ex: SGI IRIX has svc on TCP port 1)
    - Determines what svc is really listening on a port and *can even determine app name and version*
    - Operates in optional obfuscation mode
- How to detect attack?



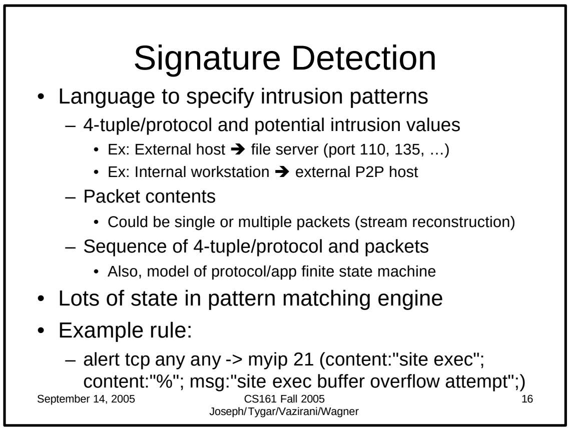
Intrusion Detection Using Signals

- This is a misuse detection problem
  - Similar problem to virus detection
  - “Match what you know”
- High-level solution:
  - Collect info about attack methods and types
    - 4-tuple/protocol
    - Packet contents
  - Create and look for signatures
    - Slammer packet, port scan, …



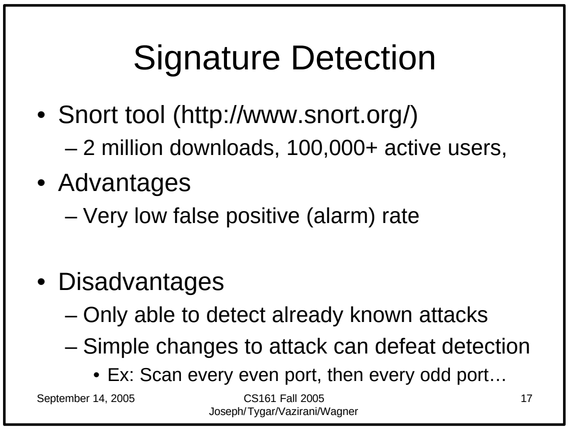
Intrusion Detection Using Noise

- This is an anomaly detection problem
  - Need to learn normal behavior
  - “Match what’s different”
- High-level solution:
  - Try to identify what is normal traffic
    - Common 4-tuple/protocol
  - Heuristic: Look for major deviations (outliers)
    - Ex: unusual target port, source addr, or port sequence (scan)
  - Apply AI: Statistical Learning Techniques



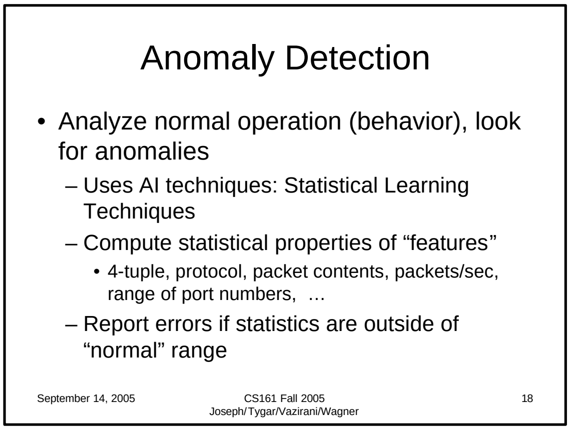
Signature Detection

- Language to specify intrusion patterns
  - 4-tuple/protocol and potential intrusion values
    - Ex: External host ➔ file server (port 110, 135, …)
    - Ex: Internal workstation ➔ external P2P host
  - Packet contents
    - Could be single or multiple packets (stream reconstruction)
  - Sequence of 4-tuple/protocol and packets
    - Also, model of protocol/app finite state machine
- Lots of state in pattern matching engine
- Example rule:
  - alert tcp any any -> myip 21 (content:"site exec"; content:"%"; msg:"site exec buffer overflow attempt";)



Signature Detection

- Snort tool (http://www.snort.org/)
  - 2 million downloads, 100,000+ active users,
- Advantages
  - Very low false positive (alarm) rate
- Disadvantages
  - Only able to detect already known attacks
  - Simple changes to attack can defeat detection
    - Ex: Scan every even port, then every odd port…



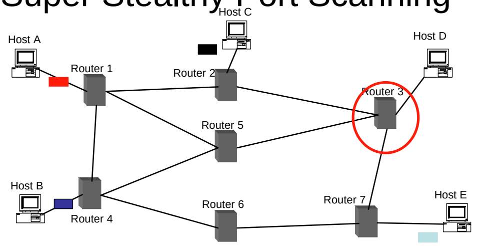
Anomaly Detection

- Analyze normal operation (behavior), look for anomalies
  - Uses AI techniques: Statistical Learning Techniques
  - Compute statistical properties of “features”
    - 4-tuple, protocol, packet contents, packets/sec, range of port numbers, …
  - Report errors if statistics are outside of “normal” range

Anomaly Detection

- Advantages
  - Can recognize “evolved” and new attacks
- Disadvantages
  - High false positive rate (alarms)
  - May have delayed alarm
  - Some attacks can hide in “normal” traffic
  - SLT requires training on known good data
  - Hard to capture protocol state behavior (FSM)
  - Problems when what’s “normal” changes
    - Ex: flash crowds

Super Stealthy Port Scanning



- Use many zombies (each scans a few ports/hour of target)
  - Each zombie is assigned many machines to scan
- Fast to scan both one machine, and many
- Very hard to detect at targets!

Distributed Intrusion Detection

- Place appliance in the network at choke point or, *share results across machines*
- Apply signature or anomaly detection across larger data set
- Advantages:
  - Easier to detect stealth probes of large number of machines
- Disadvantages:
  - Large amount of data to communicate

Honeypots

- Closely monitored network decoys
- May distract adversaries from more valuable machines on a network
- May provide early warning about new attack and exploitation trends
  - Enables in-depth examination of adversaries during and after exploitation

Honeypots

- Can simulate one or more network services on one or more machines
  - Can have virtual cluster of machines
- Causes an attacker to think you're running vulnerable services that can be used to break into the machine
  - Can log access attempts to those ports, including the attacker's source IP and keystrokes
  - Can watch attacker in real-time and trace back/forward
- Provides advanced warning of an attack
  - Could use to automate generation of new firewall rules

Tarpits

- A very,very sticky honeypot…
- Set up network decoy
  - For each port we want to “tarpit,” we allow connections to come in, but don’t let them out
- Idea:
  - Slow down scanning tools/worms to kill their performance/propagation because they rely on quick turnarounds
  - Might also give us time to protect real hosts

Example Tarpit Implementation

- Accept any incoming TCP connection
- When data transfer begins to occur, set TCP window size to zero, so no data can be transferred within the session
- Hold the connection open, and ignore any requests by remote side to close session
- Attacker must wait for the connection to timeout in order to disconnect

Tarpits

- Advantages
  - Can customize for specific worms
    - Ex: analyze incoming packets to port 80 and only tarpit web connections from worms – look for “cmd.exe” (CodeRed) or “default.ida” (Nimda)
- Disadvantages
  - Might trap valid host
  - Can cause some operating systems to crash

Intrusion Prevention Systems

- We can detect intrusions, so why not automatically cut off network connections to compromised hosts?
- Intrusion Prevention Systems do this

- But, what if we’re wrong…
  - Possible Denial of Service – trick IPS into thinking host is compromised
  - Turn off access our airline reservation server when a fare deal causes very high/different traffic patterns

Witty Worm (Mar 04): Attacking the IDS

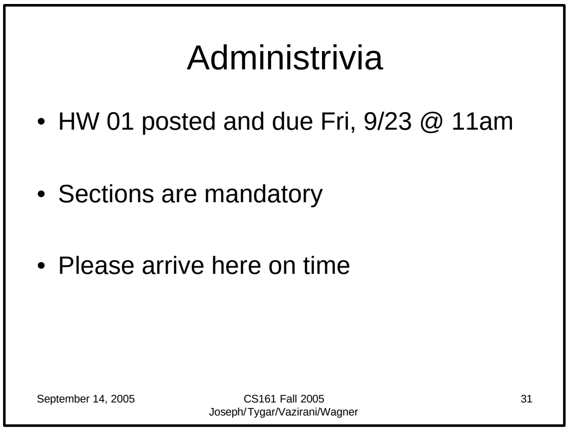
- Targeted a buffer overflow vulnerability in several of a vendor’s IDS products
- Deletes a randomly chosen sectors of hard drives over time killing system
- Payload contained phrase:
  - “(^.^) insert witty message here (^.^)”

Witty’s Many Firsts

- First widely propagated Internet worm with a destructive payload
- First worm with order of magnitude larger hit list than any previous worm
- Shortest known interval between vulnerability disclosure and worm release – 1 day
- First to spread through nodes doing something proactive to secure their computers / networks
- Spread through a population almost an order of magnitude smaller than that of previous worms

Intrusion Detection Systems Summary

- On going arms race between attackers and detection technologies
- Real challenge is false positive rate
  - Renders most IDS useless – alerts ignored
- Adaptive, anomaly detection is promising, but still lacking
- IPS products are still immature and problematic
- IDS products are now targets



Administrivia

- HW 01 posted and due Fri, 9/23 @ 11am
- Sections are mandatory
- Please arrive here on time