Baseline Intrusion Detection Evaluation
- Normal Data
- Attacks
- Evaluating Performance

Approaches to Intrusion Detection
- Bottleneck Verification to Find Illegal Root Transitions
  - Sniffer Based
  - Audit Based

Summary and Future Plans
Baseline Sniffer-Based Intrusion Detection System

- Monitors Many Workstations with one Sniffer, Computes Probability of Attack for Every TCP/IP Session
- Attack Probability Based Primarily on Number of Times Keyword Strings Occur in Session Transcripts
- Examples of Keyword Strings
  - ftp: root, anonymous, failed login (>3)
  - login: guest, root, incorrect, daemon, passwd, permission denied
- Similar to Net Ranger, and other NSM Derivative Systems
- Provide Baseline Performance Measure
Sample Transcript Showing Keywords

HP-UX hqdadev A.09.03 D 9000/750 (ttyt1)login: ~tftpPassword:Login incorrectlogin: efsPassword:Login incorrectPassword:Login incorrectPassword:/usr/efs

3:03pm up 14 days, 21:33, 1 user, load average: 0.00, 0.00
User     tty           login@  idle   JCPU   PCPU  whatefs      pty/ttyt1     3:03pm     1                w/usr/efs>

ls -al

 total 16336
 drwxr-xr-x  47 root     sys         3072 Sep 23 10:41 .
 drwxr-xr-x  47 root     sys         3072 Sep 23 10:41 ...
 drwxr-x---   3 root     sys         1024 Jul  3 14:30 Perlgrep :0: /etc/ passwd

root:*:0:3:Beginning of All Things...,,976-HPUX,:/:/bin/ksh
Outline

- Baseline Intrusion Detection Evaluation
  - Normal Data
  - Attacks
- Evaluating Performance
- Approaches to Intrusion Detection
  - Bottleneck Verification to Find Illegal Root Transitions
    - Sniffer Based
    - Audit Based
- Summary and Future Plans
NORMAL BACKGROUND DATA FROM 50 SITES (July - November 1996)

Pre-Filter Assigns Warning Values to Each Connection

Transcripts Created Only for Connections With High Warning Values

Many Connections (34.6 M), Fewer Transcripts (375,000)
Outline

· Baseline Intrusion Detection Evaluation
  ± Normal Data
  ± Attacks
  ± Evaluating Performance

· Approaches to Intrusion Detection
  ± Bottleneck Verification to Find Illegal Root Transitions
    ± Sniffer Based
    ± Audit Based

· Summary and Future Plans
### Capabilities of Attacks

<table>
<thead>
<tr>
<th>ATTACK START STATE</th>
<th>LOCAL USER</th>
<th>LOCAL ROOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>REMOTE ROOT</td>
<td>GUEST LOGIN</td>
<td>NEWS</td>
</tr>
<tr>
<td>LOCAL NET ROOT</td>
<td>IP SPOOFING</td>
<td>SNIFF</td>
</tr>
<tr>
<td>LOCAL USER</td>
<td>CRACK</td>
<td>PERL</td>
</tr>
<tr>
<td></td>
<td>STACK OVERFLOW</td>
<td>FTP CORE DUMP</td>
</tr>
</tbody>
</table>

**ATTACK GOAL**
Attack Types

- Old Attacks That Have Been Known For Many Years
  - guest login, dictionary attack, internet worm, crack, sniffer, ...

- Recent Attacks That Are Less Than a Year Old
  - buffer overflows, ip spoofing, news bug, ...

- 18 Actual Illegal Root Transitions
  - perl, loadmodule, hp vhe bug, suid root shell backdoor, telnet environment variable bug, sniffed passwords
Outline

· Baseline Intrusion Detection Evaluation
  ± Normal Data
  ± Attacks
  ± Evaluating Performance

· Approaches to Intrusion Detection
  · Bottleneck Verification to Find Illegal Root Transitions
    ± Sniffer Based
    ± Audit Based

· Summary and Future Plans
Measure Performance by Artificially Injecting Intrusions

- Truth Is Known, Assume Normal Traffic Has Few True Unknown Intrusions
- Exploits
  - OLD
  - NEW
  - REAL Attacks Found in Normal Traffic
Evaluating Intrusion Detection Systems

- Vary Threshold to Detect 80% of Attacks, Then Ignore Rejected Transcripts and Examine Accepted Transcripts
Workload for Baseline System

- Average Number of Transcripts a System Administrator Must Examine to Detect One Attack
- Set Threshold to Detect 80% of Attacks
- Workload Assumes 8 Hour Days, 1 Minute Per Transcript

<table>
<thead>
<tr>
<th>ATTACK TYPE</th>
<th>BASELINE SYSTEM</th>
<th>REAL</th>
<th>OLD</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 DAYS</td>
<td>2,800</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 DAYS</td>
<td>15,590</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90 DAYS</td>
<td>43,000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

· Baseline Intrusion Detection Evaluation
  - Normal Data
  - Attacks
  - Evaluating Performance

· Approaches to Intrusion Detection
  - Bottleneck Verification to Find Illegal Root
    - Sniffer Based
    - Audit Based

· Summary and Future Plans
Current Approaches to Intrusion Detection

- FINDS NEW ATTACKS
- ONLY FINDS OLD ATTACKS

- BOTTLENECK VERIFICATION
- SIGNATURE ANALYSIS
- SPECIFICATION-BASED
- COMPUTATION AND MEMORY REQUIREMENTS
- ANOMALY DETECTION
Outline

· Baseline Intrusion Detection Evaluation
  ± Normal Data
  ± Attacks
  ± Evaluating Performance

· Approaches to Intrusion Detection
  ± Bottleneck Verification to Find Illegal Root Transitions
    ± Sniffer Based
    ± Audit Based

· Summary and Future Plans
BOTTLENECK VERIFICATION

· DETECT IMPORTANT SYSTEM SECURITY STATE CHANGES (e.g. User Obtains Root Privilege, Log Into Remote Machine)

· DETECT USE OF LEGAL STATE CHANGE TOOL THAT ENFORCES A BOTTLENECK (e.g. Valid SU Command)

· FLAG AN ATTACK IF STATE CHANGE OCCURS WITHOUT GOING THROUGH THE BOTTLENECK

· DETECTS BOTH KNOWN AND NEW ATTACKS
Outline

· Baseline Intrusion Detection Evaluation
  ± Normal Data
  ± Attacks
  ± Evaluating Performance

· Approaches to Intrusion Detection
  ± Bottleneck Verification to Find Illegal Root Transitions
    ± Sniffer Based
    ± Audit Based

· Summary and Future Plans
Evaluating Sniffer-Based Bottleneck Verification

- Detect User Obtaining Root Privileges in an Unusual Manner
  - Does Not Detect Other Attacks (Crack, Poor Password, ...)

- Found 16 Attacks in 460 Suspicious Sessions
  - Marks Line in Transcript Where Attack Occurs
  - A Few Hours of Human Examination Found 16 Attacks
  - We Hadn't Seen Most of These Attacks Before
  - Missed Two Attacks Where the Transition to Root Was Not in the Transcript (Stolen Password, Telnet Environmental Variable Attack)

LINCOLN BOTTLENECK VERIFICATION -> HUMAN ANALYST

SUSPICIOUS SESSIONS (460) -> 16 ILLEGAL ROOT TRANSITIONS
Attacks include all illegal root incidents (18) found in four months (86,015 transcripts) of data.

- Workload reduced from 6 days per attack to 30 minutes.

Assumes 1 minute per transcript, 8 hour days.

**BASELINE BOTTLENECK**

**VERIFICATION**

**REAL ATTACKS**

- 2,800 Transcripts
  - 6 days
- 29 Transcripts
  - 30 minutes
EXAMINE PERL ATTACK SCRIPT

MAKE SET-ID SCRIPT

RUN SCRIPT

USER IS NOW ROOT!

TRY TO COVER TRACKS

WHO'S WATCHING?
Why Bottleneck Verification Performance Works so Well

1) Parses User Interaction and Exploits Local Context
   - Login Sequence, Banner, Shell Prompt, Command, Command Options, and Command Outputs Are Processed Separately

2) Minimize False Alarm Rates
   - Detect Only Breaking Actions
   - Most Signature Analysis Systems Do Not Select Strings to Minimize False Alarms

3) Process Only Telnet Shell Sessions
   - Many Air Force Sessions Are Special Applications, Not Shell Sessions
Outline

1. Baseline Intrusion Detection Evaluation
   - Normal Data
   - Attacks
   - Evaluating Performance

2. Approaches to Intrusion Detection
   - Bottleneck Verification to Find Illegal Root
     - Sniffer Based
     - Audit Based

3. Summary and Future Plans
Host Based Intrusion Detection System

- Uses Audit Data From Sun's Basic Security Module (BSM)
- Looks for a Privileged Shell Launched From a Non-Privileged User Shell Without an Su
- Low-Complexity Real-Time Implementation
- Successfully Finds Stealthy Buffer Overflow Illegal Root Transitions
Summary

A Simple Sniffer-Based Baseline Intrusion Detection System Requires High Estimated Workloads

Sniffer-Based Bottleneck Verification ± Reduces Estimated Human Workload by Two Orders of Magnitude and Successfully Finds Unknown Attacks

Audit-Based Bottleneck Verification ± Allows Real-Time Low-Complexity Operation and Finds Stealthy Attacks that Would be Missed by Sniffer

Planning Further Evaluations and Extensions ± Improve Sniffer Script Parsing ± Detect More Complex Attacks with Audit Data ± Evaluate Using DARPA Intrusion Detection Evaluation Training/Test Data Being Created at Lincoln Laboratory