Designing and Implementing Malicious Hardware

Samuel T. King, Joseph Tucek, Anthony Cozzie, Chris Grier, Weihang Jiang, and Yuanyuan Zhou

Andrew De Zeeuw
Matthew Lindstrom
Outline

● Motivation
● Goal
● Approach
  ○ General Purpose Hardware
  ○ Implemented Attacks
● Results/Overhead
● Possible Defense
● Discussion Points
Motivation

● DoD Report
  ○ High performance ICs are expensive to produce
  ○ IC production limited to consumer suppliers
  ○ Consumer suppliers are moving design, manufacturing, and testing to other countries

● Software / Hardware Hacks
  ○ Apple - RavMonE virus
  ○ IBM Trojan Circuit
Goal

- Discuss locations for potential circuit level attacks
- Design and Implement potential attacks
- Promote future research for malicious hardware attacks
Approaches

● General purpose hardware:
  ○ Memory Access
  ○ Shadow Mode

● Implemented attacks:
  ○ Privilege Escalation
  ○ Login Backdoor
  ○ Stealing Passwords
Memory Access

- State machine monitors data bus for a magic value enabling the attack
- Memory management unit allows access to privileged data for attackers
Privilege Escalation Attack

- Escalate user to root using memory access
- Search protected kernel memory and elevate effective user to root
Shadow Mode

- Instructions have full processor privilege and are invisible to the system
- Instruction and Data cache lines reserved for attack
  - Timing perturbed during attack
- Two Bootstrap Mechanisms
  - Bootstrap Code / Bootstrap Trigger
Login Backdoor Attack

1. Attacker sends unsolicited UDP packet

2. Monitor notices the magic byte sequence

3. Firmware is copied to reserved cache area and activated.

4. 
   
   ```
   login: root
   password: letmein
   
   Last login: Mon Apr 1 [root@victim ~]
   ```

   Attacker logs in as root. Shadow firmware uninstalls.
Stealing Passwords Attack

- Monitor write system call for “Password:” string
- Grab corresponding read value
- Store them in shadow memory
- Send passwords using one of two options:
  - System calls to network interface
  - Overwrite UDP packet with predetermined IP
Circuit-Level Implementation

- FPGA with a modified Leon3 processor
- Memory Access Modifications
  - Data Cache
  - Memory Management Unit
- Shadow Mode Modifications
  - Instruction and Data Cache
  - Watchpoint to trap load/store values (i.e. passwords)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline CPU</td>
<td>1,787,958</td>
<td>11,195</td>
<td>~</td>
<td>~</td>
</tr>
<tr>
<td>CPU + Memory Access</td>
<td>1,788,917</td>
<td>11,263</td>
<td>959</td>
<td>68</td>
</tr>
<tr>
<td>CPU + Shadow Mode</td>
<td>1,789,299</td>
<td>11,312</td>
<td>1,341</td>
<td>117</td>
</tr>
</tbody>
</table>
Performance - Login Backdoor

Timing Perturbations

<table>
<thead>
<tr>
<th>Program</th>
<th>Baseline</th>
<th>Known Root</th>
<th>Transient</th>
<th>Persistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcc</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parser</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>twolf</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bzip2</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wget</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AVG	| Overhead |
---|----------|
Known Root	| 1.32%    |
Transient	| 1.34%    |
Persistent	| 13.00%   |
Possible Defenses

- Defenses against malicious hardware developed after this paper was published
- Detection of malicious hardware using analog and digital perturbations introduced by attack
  - Analog
    - Side-channel analysis to detect
    - Difficult to hide signal distortions due to malicious hardware
    - However side-channel analysis started as an attack
  - Digital
    - Digital testing may fail to detect hardware
    - Reverse engineering complete IC
    - Redundant hardware
Conclusion

- Designed two general purpose malicious hardware circuits
- Implemented three attacks with low area overhead and timing cost
- Further research in malicious hardware is necessary to prevent future attacks
Discussion Points

● Hardware attacks are hard coded and therefore single purposed, discuss the pros and cons of this attack.
● Hardware attacks are expensive to implement and defend against, for what applications might this attack be best suited?
● Are there any simple techniques the OS could use to nullify the attacks listed in this paper?
● How feasible is it to maliciously insert hardware?