CSE331 Computer Security Fundamentals

Covert channels

Nick Nikiforakis
nick@cs.stonybrook.edu
Covert channels?

• In the previous lecture we talked about covert channels, and how they can be used to bypass the security guarantees of the BLP model.

• **Covert channel**: communication channel based on the use of system resources not normally intended for communication between the subjects (processes) in the system.
Covert channels

- Covert channels are used as a means of communication between two processes.
- The processes may be:
  - Authorized to communicate, but not in the way they actually are
  - Prohibited from communicating
- One process is a Trojan
  - Transmits data in a covert manner
- The other process is a Spy
  - Receives the data sent by the Trojan
Where and What?

• For a covert channel to exist, it must be the case that:
  – A multi-level system is in use
  – A resource (or one of its attributes) is shared by high (Trojan) and low (spy) processes

• Types of channels:
  – Storage channel
    • Data stored by one process to be read by another
  – Timing channel
    • Some system parameter is modulated
Definitions

• *Covert channel*: Intentionally used to communicate

• Later in the course, we may get a chance to briefly review:
  – *Side channels*: Unintentionally reveal information
  – *Steganography*: Techniques for hiding messages within other messages. Sometimes used to hide the presence of communication
    • [http://www.garykessler.net/library/steganography.html](http://www.garykessler.net/library/steganography.html)
(Why) Are They Important?

- Difficult to detect
  - The channels that are abused for communication are typically overlooked by system administrators and access-control systems
- Can operate for a long time and leak a substantial amount of classified data to uncleared processes
  - Trojan/Spy scenario
- Can compromise an otherwise secure system, including one that has been formally verified!
- Must be considered to achieve high government certification levels
Local Channels
Resource Manipulation

• Trojan fills kernel’s process table to transmit 1, leaves it partially empty to transmit 0. Spy tries to create process.
  – If it succeeds, writes down 0, else 1.
• Trojan allocates 0MB of memory to transmit 00, 64MB to transmit 01, 128MB to transmit 10, 192MB to transmit 11.
  – Easily distinguishable by any spy with resource monitoring capabilities
• Trojan induces bus contention, spy measures bus latency (multiprocessors)
  – These things apply on the cloud as well
• In multicore systems, execute CPU-intense program for 1, else 0
Disk Arm Optimizations

Attack

To send a bit:
Low: 2
To send a 0: High: 1
To send a 1: High: 3
Low: 0, 4

Spy process observes which request finishes first to receive bit:
- 0 first = 0 transmitted
- 4 first = 1 transmitted

Karger, Wray, “Storage Channels in Disk Arm Optimization”
Resource Exhaustion Countermeasures

• Pre-allocate resources and prevent dynamic modification
  – Partial solution only, and not always applicable
• Put quotas on processes and users
  – E.g. user A cannot run more than N processes
• Only used when covert channels pose a serious enough risk to justify the inefficiency
Criteria for the presence of covert channels

**Covert Storage Channel**
- Sender and receiver have access to some attribute of a shared object
- Sender must be able to modify the attribute
- Receiver must be able to view that attribute
- There must be a mechanism for initiating both processes and sequencing their actions to the shared resource

**Covert Timing Channel**
- Sender and receiver have access to some attribute of a shared object
- Both have access to a time reference (e.g. real-time clock)
- The sender must be able to control the timing of the detection of a change in the shared attribute
- There must be a mechanism for initiating both processes and sequencing their actions to the shared resource
General Characteristics

• Characteristics of a covert channel:
  – Existence: Is there a channel present that can be abused?
  – Bandwidth: How many bits per second can be transmitted?
  – Noise-levels: How much noise is there on this channel?

• Once we find one:
  – Eliminate it
  – Reduce Bandwidth / Increase Noise
  – Monitor for exploitation
Possible defense: Fuzzy Time

- All covert **timing** channels rely on accurate clock
- You can either attempt to disrupt the timing of the channel (add noise or slow it down), or reduce the accuracy of the clock
- VAX security kernel slows down timer interrupt periods to be uniformly distributed with a mean of 20 ms.
- Randomly modifies the completion time of I/O requests, so they can’t be used as a clock

Network/Remote Channels
Network channels

- **Definition:** Mechanisms for sending information without the knowledge of the network administrator or other users.

- **Key idea:** Exploit new/existing outgoing communication channel to covertly leak information

- The Trojan is local but the Spy is remote

- As with local channels, there is a very wide range of available techniques
Why?

• Purpose of network covert channels:
  – Exfiltrate data without alerting network administrators
    • E.g. Using FTP/SCP to copy files to previously unseen remote hosts may trigger an alert
  – Malware can receive commands from a Command-and-Control server
  – Bypass local firewalls
    • E.g. my work does not allow me to use Facebook
  – Bypass censorship
    • E.g. oppressive governments limit access to certain types of information
How?

• Identify network protocols with fields that allow the client and the server to exchange arbitrary data
  – DNS
  – ICMP
  – TCP
• Encode data according to the protocol that one wants to abuse
  – E.g. Binary vs Base32/Base64
• Incorporate the encoded data in outgoing traffic
• Decode covertly-sent data from incoming traffic
Figure 1. The IP protocol header

Normally used for fragmentation control

Murdoch, Lewis 2005, “Embedding Covert Channels into TCP/IP”
TCP Channels

<table>
<thead>
<tr>
<th>Source Port</th>
<th>Destination Port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Acknowledgment Number</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>HLEN</td>
<td>Reserved</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Options</td>
<td>Padding</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2. The TCP protocol header
Packet one:
537657344:537657344(0) win 512 (ttl 64, id 18432)

Decoding… (ttl 64, id 18432/256) gives ASCII 72 (H)

Packet two:
537657344:537657344(0) win 512 (ttl 64, id 17664)

Decoding… (ttl 64, id 17664/256) gives ASCII 69 (E)

Packet three:
537657344:537657344(0) win 512 (ttl 64, id 19456)

Decoding… (ttl 64, id 19456/256) gives ASCII 76 (L)
Packet four:
537657344:537657344(0) win 512 (ttl 64, id 19456)

Decoding… (ttl 64, id 19456/256) gives ASCII 76 (L)

Packet five:
537657344:537657344(0) win 512 (ttl 64, id 19456)

Decoding… (ttl 64, id 19456/256) gives ASCII 79(O)

Packet six:
537657344:537657344(0) win 512 (ttl 64, id 2560)

Decoding… (ttl 64, id 2560/256) gives ASCII 10 (carriage return)
ICMP Channels

- ICMP echo request/reply can tunnel arbitrary user data
  - Payload capacity depends on path MTU (this feature often used to measure PMTU)

Sohn, Noh, Moon 2003, “Support Vector Machine Based ICMP Covert Channel Attack Detection”
HTTP Channels (legitimate!)

- SOAP messages (web services) use covert channels: HTTP tunnels
- HTTP tunneling supported by almost all applications that wish to circumvent firewalls
  - Instant messaging
  - Blocked websites
  - Applications that require specific blocked ports
  - ...

DNS Channels

- DNS can hold arbitrary text in its various fields
- Software:
  - OzyManDNS
  - Iodine
- Uses
  - Bypass firewalls
  - Bypass payment step in paid WiFi zones (airports, airplanes?, cafes, etc.)
How do we detect network covert channels?

• Intrusion detection systems
  – Use signatures to detect tell-tale signs of protocols being tunneled
    • E.g. Ready-made software for network covert channels use specific headers that can be encoded into signatures
  – Anomaly detection
    • Sudden spikes in volume of ICMP/DNS requests
    • Unusually large ICMP responses
    • Random-looking subdomains in DNS resolutions
Summary

• Covert channels vs overt channels
  – Storage
  – Timing

• Hard to protect against them in real systems

• Not as important in non-military systems (systems with DAC)
  – Network channels, however, are important for evading data exfiltration detection of modern IDSs

• Elimination/Reduce Bandwidth/Monitoring