THE FUTURE OF DATA EXFILTRATION & MALICIOUS COMMUNICATION

Steffen Wendzel
http://www.wendzel.de
• Steffen Wendzel
  • PhD student @University of Hagen
  • Researcher @Augsburg University of Applied Sciences
  • Author of four CS-related books

http://www.wendzel.de  | Twitter: @cdp_xe
Prediction

Malware communication will become *stealthy* and *adaptive*.
Requirement

Hide communication between sender and receiver, i.e., provide a communication that raises as few attention as possible

… can be used by journalists to transfer illicit information but also by malware
Part I

The hiding techniques we already know …

… and what research did to counter network covert channels.
Typical Techniques for Covert Channels

- Packet Timings
- Packet Order
- Find something to piggyback (unused/redundant fields in ICMP, HTTP, etc.)
Typical Techniques for Covert Channels

- Many of the available hiding techniques & programs implement *crapto channels*.

*adopted from „craptography“, i.e., crappy crypto implementations.
# Shared Resource Matrix

<table>
<thead>
<tr>
<th>Attribut</th>
<th>Operation A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Op1</td>
</tr>
<tr>
<td>a</td>
<td>R</td>
</tr>
<tr>
<td>b</td>
<td>-</td>
</tr>
<tr>
<td>c</td>
<td>-</td>
</tr>
<tr>
<td><strong>User-In</strong></td>
<td>R</td>
</tr>
<tr>
<td><strong>User-Out</strong></td>
<td>M</td>
</tr>
</tbody>
</table>
Covert Flow Trees

procedure Lockfile(f: file);
begin
  if not f.locked and empty(f.inuse) then
    f.locked := true
end;

procedure Unlockfile(f: file);
begin
  if f.locked then
    f.locked := false
end;

function Filelocked(f: file): boolean;
begin
  Filelocked := f.locked;
end;

<table>
<thead>
<tr>
<th></th>
<th>Lockfile</th>
<th>Unlockfile</th>
<th>Filelocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>locked,inuse</td>
<td>locked</td>
<td>locked</td>
</tr>
<tr>
<td>modify</td>
<td>locked</td>
<td>locked</td>
<td>-</td>
</tr>
<tr>
<td>return</td>
<td>-</td>
<td>-</td>
<td>locked</td>
</tr>
</tbody>
</table>
Covert Flow Trees

<table>
<thead>
<tr>
<th></th>
<th>Lockfile</th>
<th>Unlockfile</th>
<th>Filelocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference</td>
<td>locked,inuse</td>
<td>locked</td>
<td>locked</td>
</tr>
<tr>
<td>modify</td>
<td>locked</td>
<td>locked</td>
<td>-</td>
</tr>
<tr>
<td>return</td>
<td>-</td>
<td>-</td>
<td>locked</td>
</tr>
</tbody>
</table>

Covert storage channel via attribute locked

Modification of attribute locked

Recognition of attribute locked

Direct recognition of attribute locked

Indirect recognition of attribute locked

Filelocked

Infer attribute locked via attribute inuse

Image Source: M. Bishop: "Computer Security. Art and Science"
The Pump and Similar Approaches
Traffic Normalization

- Clear/Unify/Modify selected areas in network packet headers
- Cold Start Problem
- Inconsistent TCP retransmissions

Image source: Handley et al., 2001
Fuzzy Time

- 1991 (VAX Security Kernel)
Other Approaches

- Statistical approaches
- Machine learning
- Various active wardens
- Business process evaluation
- Spurious processes approach
- Code modifications to prevent covert channels based on timing leaks
- ... and quite many other academic approaches (cf. my latest book)
Summary (pt. 1)

Many means exist to

... embed hidden information into network packets

... to detect, limit, and prevent such embeddings

... some of them are ooold!

... but we cannot detect all techniques.
Part II

Novel Approaches for Network Covert Storage Channels

[selected aspects of a thesis]
Related Work

- Existing CC-internal Control Protocols (Ray/Mishra, pingtunnel)
- Natural Selection for Network Protocols (Li et al.)
- Adaptive Network Covert Channels (Yarochkin et al.)
- Covert channels optimized for raising low attention using CC-internal Control Protocols
  ... and Protocol Hopping Covert Channels
  ... able to bypass normalizers.
- Protocol Channels
Features

- Protocol Switching
  - Adaptive Covert Channels
  - Network Environment Learning Phase (NEL)
  - Mobile Environments
- Version-dependent protocol sets
  - Step-by-step Upgradability
- Space-efficiency and dynamic headers
Terminology

- **Terminology** as a means to provide finer distinctions between different points of view.

- **Underlying Protocol**
  - e.g. IPv4, TCP, ICMPv4, IPv6, …

- **Cover Protocol**
  - utilized area within the underlying protocol
  - e.g., 2 least significant bits of TTL + DF flag

- **Micro Protocol**
  - control protocol placed within cover protocol
  - shares cover protocol space with the covert channel's payload
Combining Multiple Layers

\[ s_{pkt}(\text{IMAP}) + s_{pkt}(\text{TCP}) + s_{pkt}(\text{IP}) + s_{pkt}(\text{Ethernet}) \]
Micro Prot. Engineering Approach

Layer 2: Micro Protocol

3 micro protocol design
output: header bits of the micro protocol

4 evaluate micro protocol bits
output: probability list for micro protocol's bits

Layer 1: Underlying Network Protocol

1 define cover protocol
output: selection of underlying protocol's bits to utilize (=cover protocol)

2 evaluate cover protocol bits
output: probability list for cover protocol's bits

5 list mapping
output: mapped lists

6 verify reachability
output: underlying protocol conform micro protocol

$L_{MP} = p_{40}, p_{31}, p_{41}, ...$
$L_{CP} = p_{10}, p_{21}, p_{11}, ...$
Status Update Approach

- We tried to adopt existing protocol engineering means
- IPv6 „Next Header“, IP „Options“
- Compressed SLIP (CSLIP)
- Status Updates are is like a mix of „Next Header“, „IP Options“, and „CSLIP“.
Status Updates

- We link a communication between two CC peers to *statuses*.
- A connection can comprise different statuses, e.g.:
  - Source address
  - Destination address
  - Transaction state
- Status Updates indicate the update of a status.
Status Updates

- One status update comprises
  - A „Type of Update“ value
  - The value for the update
- Therefore, sender and receiver share a ToU table, e.g.:
  - 00 SET SOURCE ADDRESS
  - 01 SET DESTINATION ADDRESS
  - 10 END OF UPDATES
  - 11 PAYLOAD follows directly
## Status Updates

- For instance, to change the source address of a connection (e.g., on a proxy):

| 00 (SET SRC ADDR.) | NEW SOURCE ADDRESS FOR THE CONNECTION  
|--------------------|----------------------------------------|
|                    | (e.g., a small n bit overlay address or 
|                    | an underlay network's address)          |
Example: Packet Forwarding

Sender

Peer x

Peer y

SET DESTINATION ADDRESS=y

PAYLOAD Follows
DIRECTLY="ABC"

PAYLOAD Follows
DIRECTLY="DEF"

SET DESTINATION ADDRESS=x

PAYLOAD Follows
DIRECTLY="ABC"

PAYLOAD Follows
DIRECTLY="DEF"
Combining ToUs to Sequences

<table>
<thead>
<tr>
<th></th>
<th>New Source Address</th>
<th>01</th>
<th>New Destination Address</th>
<th>10</th>
<th>/ unused /</th>
</tr>
</thead>
</table>
Re-Design of Ray/Mishra'08

- Designed a status update-based version of a CC micro protocol developed by Ray and Mishra.

a) unmodified header (8 bits):

<table>
<thead>
<tr>
<th>seq. number</th>
<th>data flag</th>
<th>ack flag</th>
<th>exp. seq. no.</th>
<th>start flag</th>
<th>end flag</th>
</tr>
</thead>
</table>

b) re-designed header, default ToU (7 bits):

<table>
<thead>
<tr>
<th>ToU</th>
<th>seq. number</th>
<th>data flag</th>
<th>ack flag</th>
<th>exp. seq. no.</th>
</tr>
</thead>
</table>

c) re-designed header, start/stop ToU (3 bits):

<table>
<thead>
<tr>
<th>ToU</th>
<th>start flag</th>
<th>end flag</th>
</tr>
</thead>
</table>
Re-Design of Ray/Mishra'08

covert channel protocol header size

summarized header size (in bits)

sent packets

+ protocol by Ray/Mishra
- status updates
+ huffman coding
Re-Design of Ray/Mishra'08

- Initial connection inefficiency problem
  - Many ToUs are required to initially configure a connection
  - These ToUs require more space than a normal header
  - SU perform better if a transaction requires $\geq 5$ packets
Dynamic Routing in CC Overlays

- CC networks are overlay networks
- Similar to Ad-Hoc networks (changing components, changing topology)
- Existing approach for dynamic routing in steganographic networks was presented by Szczypiorski et al. and was based on the random-walk algorithm.
Requirements for CC Routing

- Routing overhead should be small
  - Status updates
- Must be capable to adapt quickly to topology changes since underlay network can change at any time.
  - Only small routing overhead should be produced for propagating updates.
- Overlay network addresses can differ to underlay addresses and a routing approach must support overlay addresses.
Our Approach

• Sender is responsible for route plotting (source routing).
• We implemented **optimized link state routing** (OLSR)
  • OLSR was designed for mobile Ad-Hoc networks
  • … with the goal of a small routing overhead
  • Multi-Point Relays (MPR)
    – A peer floods updates only to peers of his MPR set (similar like OLSR).
• Status Update-based realization to achieve minimal micro protocol overhead
Dynamic Routing in CC Overlays

• Introducing **Quality of Covertness**
• Extendable Architecture
• Dynamic **Cover** Protocol Switching
  • Protocol Hopping Covert Channels
• Network Environment Learning Phase
  • Peers determine possible communication options between each other
Agents and Drones for Overlay Routing

Drones do not take part on routing decisions and are never a routing path's destination.

Drones are not aware of a covert communication.
Agents and Drones for Overlay Routing

- Our approach comprises a CC network topology table
  - A graph of the paths between peers as well as their capabilities (supported CC techniques)
  - Is propagated between the peers
- New ToUs for routing propagation were required:

<table>
<thead>
<tr>
<th>Type of Update</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUEST_PT_TT</td>
<td>Used by a peer to request the full peer table and topology table while bootstrapping.</td>
</tr>
<tr>
<td>RESPONSE_PT_TT</td>
<td>Response to REQUEST_PT_TT.</td>
</tr>
<tr>
<td>TT_LIST</td>
<td>A sequence of edges of the topology graph. Send on topology changes. Propagated according to MPRsel.</td>
</tr>
<tr>
<td>PT_ENTRY</td>
<td>A new or updated entry to the peer table. Send when a peer crashes, goes off, or joins the network, or changes CC capabilities. Propagated according to MPRsel.</td>
</tr>
</tbody>
</table>
Agent Clusters
SCCT
What can we do to counter PSCCCs?

- By introducing delays on protocol switches
- PoC code based on delay-net/IPQueue and iptables
Example

- Protocol Channel based on ICMP (1) & UDP (0)
- Message „0110001“ with high \(d\) (e.g. 1s)

Active Warden Input:

Output: U,I,U,U,I,U,I or 0100101
Results

- Pretty useful to counter protocol channels!
- Can counter protocol hopping covert channels without sequence numbers in their micro protocols!
Summary (pt. 2)

- Improved CCs with protocol hopping
- CC overlays with dynamic routing capability
  - Agents and Drones
  - Upgradable Infrastructure
  - Mobile Access
- Internal control protocols (micro protocols)
  - Optimized for a low-attention raising operation
  - Utilization of multiple layers for cover protocols
- Active warden to counter protocol switches
Part III

Covert and Side Channels in Building Automation Systems
Side Channels in BAS

- **Side channels** are covert channels *without intentional sender*
- A side channel in a BAS leaks information about *events* taking place within a building
- Examples:
  - Employee uses a side channel to detect the presence of his boss in his office in order to steal a document.
  - Observing healthiness / Ambient Assisted Living
Covert Channels in BAS

Enterprise network could be highly protected

→ data leakage will be difficult

Solution:

Exfiltrate confidential information using a **covert channel** (e.g., BAS broadcasting).

The receiver can either be connected to the BA network or can sniff a tunneled BA connection between multiple buildings.

- e.g., BACnet/IP (encapsulated in UDP)
BACnet Protection

Response(TempA=21°C)
Request(TempA)
Write-Down
Read-Up

High

Low
Introducing MLS using the Open Source BACnet Firewall Router

Primary BFR

Secondary BFR

Network with Devices of (secret, {sales})

Secondary BFR

Network with Devices of (secret, {r&d, project-x})

Secondary BFR

Network with Devices of (confidential, {project-x})
MLS+BFR = Protection!

BACnet/IP Device
- turn on heating on secret device

BACnet/IP Device
- read temperature from top secret device

Top Secret BFR

First Level BFR

Secret BFR
Summary (pt. 3)

- We presented the first side channels and covert channels in BAS, and especially in BACnet.
- We presented a means to protect BACnet environments based on the BACnet Firewall Router.
  - not really stable,
  - bad documentation,
  - over-engineered (configurable via „stacks“).
- We need a stable and usable BACnet firewall!
  - Any volunteers?
What can we conclude?
There are various means to establish covert channels and various (theoretical) means to counter covert channels.
Novel approaches enable covert channels to become pretty valuable for malware …

… but *should* become valuable for the „good guys“.
Covert (and Side) Channels exist in Building Automation Systems …

… but can be prevented.
However, the important thing is …
You can

... enable covert channels to become useful in practice (journalists).
... create real systems to counter the botnets of the future.
Related Publications

- **Books:**
  - Steffen Wendzel: Tunnel und verdeckte Kanäle im Netz, Springer-Vieweg, 2012. (in German)

- **Scientific Papers (Selection):**

- **Professional Articles:**