

15-412 Project: VMware Guest Tools for plan9

Thomas Tuttle

September 30, 2009

Outline

- 1 What?
- 2 Why?
- 3 How?
- 4 When?
- 5 Huh?

What?

- You should all know what **Plan 9** is by now.
- **VMware** is a popular commercial hypervisor. (There are several different products for different purposes.)
- The **VMware guest tools** allow the guest OS to communicate with the host to do useful things, such as:
 - Absolute mouse positioning
 - Clipboard sharing
 - Memory management
 - Drag and drop file transfer (scary!)

What?

- You should all know what **Plan 9** is by now.
- **VMware** is a popular commercial hypervisor. (There are several different products for different purposes.)
- The **VMware guest tools** allow the guest OS to communicate with the host to do useful things, such as:
 - Absolute mouse positioning
 - Clipboard sharing
 - Memory management
 - Drag and drop file transfer (scary!)

What?

- You should all know what **Plan 9** is by now.
- **VMware** is a popular commercial hypervisor. (There are several different products for different purposes.)
- The **VMware guest tools** allow the guest OS to communicate with the host to do useful things, such as:
 - Absolute mouse positioning
 - Clipboard sharing
 - Memory management
 - Drag and drop file transfer (scary!)

Why?

The current VMware tools for Plan 9 are:

- old

...the mechanism for backdoor calls into VMware has completely changed at least twice since I wrote `aux/vmware`, which is why it no longer works very well.

- not open-source

In order to get the information that I used to write `aux/vmware`, I signed an NDA that prohibits me from releasing the source code, regardless of what VMware has done in the interim.

Why?

The current VMware tools for Plan 9 are:

- **old**

...the mechanism for backdoor calls into VMware has completely changed at least twice since I wrote `aux/vmware`, which is why it no longer works very well.

- **not open-source**

In order to get the information that I used to write `aux/vmware`, I signed an NDA that prohibits me from releasing the source code, regardless of what VMware has done in the interim.

Why?

Why do we care about VMware tools at all?

- Good VMware support under Plan 9 makes it possible to provide a solid virtual machine image for new users to try Plan 9, potentially gaining more cultists users.
- Makes it easier for developers to work on Plan 9 under VMware. (Some devs may not care, but it's still useful to have.)

Why?

Why do we care about VMware tools at all?

- Good VMware support under Plan 9 makes it possible to provide a solid virtual machine image for new users to try Plan 9, potentially gaining more cultists users.
- Makes it easier for developers to work on Plan 9 under VMware. (Some devs may not care, but it's still useful to have.)

How?

From the VMware side:

- 1 Two assembly backdoors
 - Original, simple backdoor (only send/receive registers)
 - High-bandwidth backdoor for sending arbitrary amounts of data
- 2 Message channel
 - Layered on top of backdoor calls
 - Channels identified by cookies
- 3 RPC
 - Layered on top of message channels
 - Used for more complicated services
 - Does not appear to have been implemented in old Plan 9 tools

How?

From the VMware side:

- 1 Two assembly backdoors
 - Original, simple backdoor (only send/receive registers)
 - High-bandwidth backdoor for sending arbitrary amounts of data
- 2 Message channel
 - Layered on top of backdoor calls
 - Channels identified by cookies
- 3 RPC
 - Layered on top of message channels
 - Used for more complicated services
 - Does not appear to have been implemented in old Plan 9 tools

How?

From the VMware side:

- 1 Two assembly backdoors
 - Original, simple backdoor (only send/receive registers)
 - High-bandwidth backdoor for sending arbitrary amounts of data
- 2 Message channel
 - Layered on top of backdoor calls
 - Channels identified by cookies
- 3 RPC
 - Layered on top of message channels
 - Used for more complicated services
 - Does not appear to have been implemented in old Plan 9 tools

How?

From the Plan 9 side:

- `bin/aux/vmware` **provided** `/mnt/vmware/*`
For example, `/mnt/vmware/snarf` **allows copy/paste**
(read = paste, write = copy).
- These files mirror the standard Plan 9 interfaces for the devices.

When?

Tentative plan:

- 1 Reimplement `backdoor.*` and `msgchan.c`, so existing interfaces work.
- 2 Implement `rpc` on top of `msgchan`.
- 3 Rewrite services that have newer RPC-based interfaces (e.g. `snarf`) to it.
- 4 Implement new services, as time permits.

When?

Tentative plan:

- 1 Reimplement `backdoor.*` and `msgchan.c`, so existing interfaces work.
- 2 Implement `rpc` on top of `msgchan`.
- 3 Rewrite services that have newer RPC-based interfaces (e.g. `snarf`) to it.
- 4 Implement new services, as time permits.

When?

Tentative plan:

- 1 Reimplement `backdoor.*` and `msgchan.c`, so existing interfaces work.
- 2 Implement `rpc` on top of `msgchan`.
- 3 Rewrite services that have newer RPC-based interfaces (e.g. `snarf`) to it.
- 4 Implement new services, as time permits.

When?

Tentative plan:

- 1 Reimplement `backdoor.*` and `msgchan.c`, so existing interfaces work.
- 2 Implement `rpc` on top of `msgchan`.
- 3 Rewrite services that have newer RPC-based interfaces (e.g. `snarf`) to it.
- 4 Implement new services, as time permits.

What?

Why?

How?

When?

Huh?

Huh?

Questions?