# Configuring Bro

Seth Hall
International Computer Science Institute

```
const a_setting = T &redef;
```

```
redef a_setting = F;
```

Change settings only at startup

But this is so painful for some settings!

# Previous Solution

- **broctl update**
  - Works by sending updated redef-able consts through communication.
  - Flaky!
  - Not well supported and lots of edge cases
  - (secret, hidden feature: you can update code at runtime in very narrow cases, but that “feature” is going away)

# New Solution

- **Config framework!**
  - base/frameworks/config

# This Came From New Capabilities!

- **Now**: Input framework
- **Future**: Broker key-value store

# Changes Required

- What was **const** must now change to **global**.

```
-    const local_nets: set[subnet] &redef;
+    global local_nets: set[subnet];
```

- You use the config framework by handling an event that tells you a key has been updated.

```
1  event Config::key_update(key: string)
2      {
3      if ( key == "bro.site.local_nets" )
4          Site::local_nets = Config::get_subnets("bro.site.local_nets");
5      }
```

# What is Config::get_subnets???

- Bro's type system limits data conversion so we have to be careful how we do it.
- There are a whole set of functions which convert data from the config store into the correct Bro type.
- API is not set in stone yet.  Planned for inclusion into 2.5

# There will still be edge cases!

- Some things are only settable at start up time and can’t be modified.
- Some logging framework settings are only available at start up.
- Code cannot be updated at runtime.
    - Scripts and signatures cannot be loaded or unloaded at runtime.

# What do we gain?

- Retain flow state!
- Zero downtime for config changes!
- Migration of configuration state out of Bro scripts.

# Demo

# Questions?