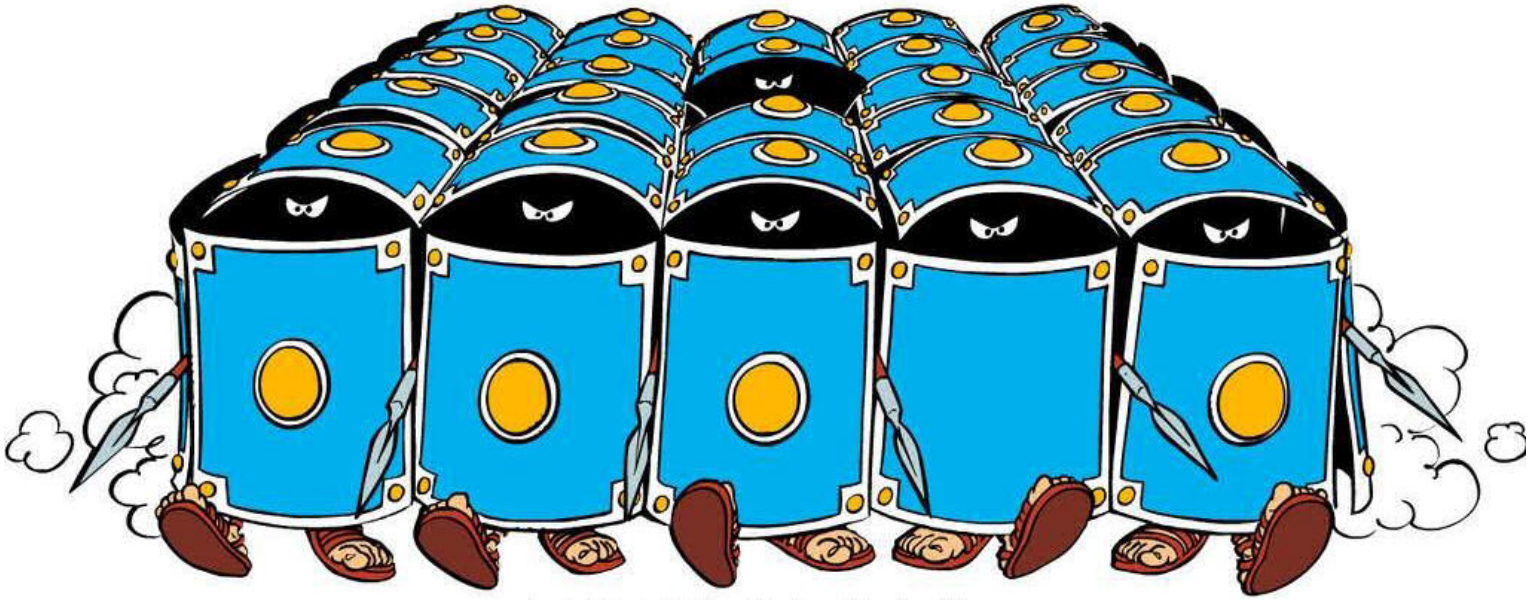
REX vs The Romans



fG! @ WhiskeyCon - SyScan'14

(Political) DISCLAIMER!

- Bad guys exist in the world, I am not against spying and busting them.
- The problem is that it is too easy and attractive to abuse power.
- History repeats itself all the time.
- The process is not transparent so it can be (and it is) abused.

Who Am I?

- Someone bleached my hat and now I am a whitehat.
- Trying to make your (enterprise) Macs (more) secure (don't laugh!).
- Bla bla bla bla bla…

Hacking Team

Hacking Team

- ❑ Lame?
- ❑ Dumb?
- ❑ Skill-less?
- ❑ Greedy?
- ❑ Unable to write a rootkit?
- ❑ Unable to write a packer?
- ❑ All of the above?

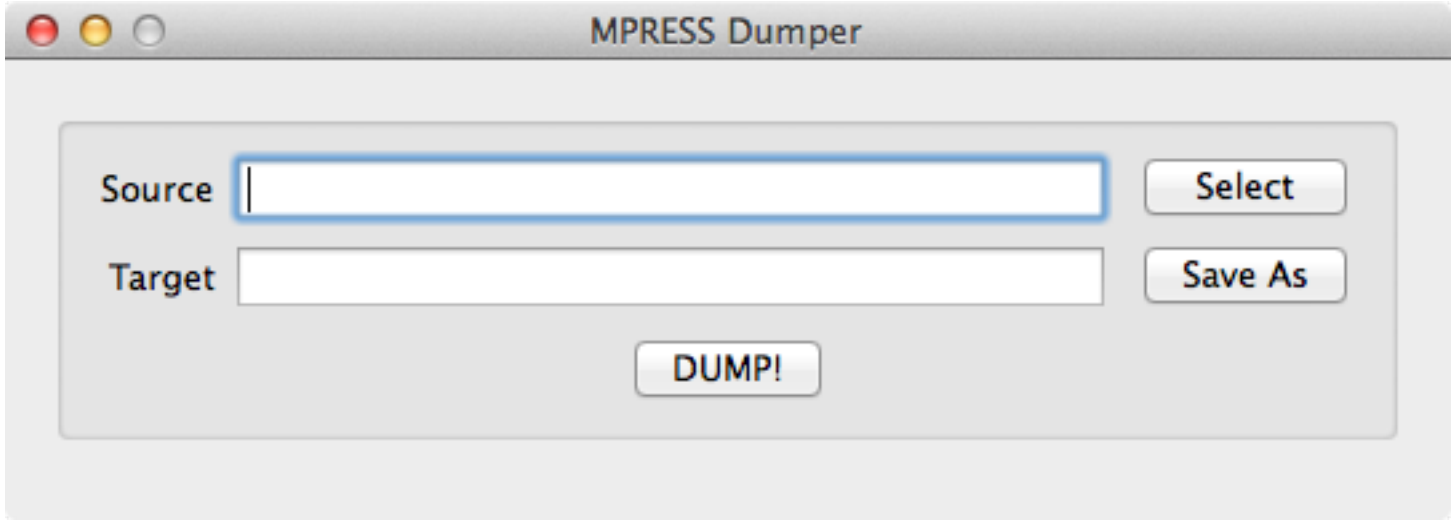
The dropper

The dropper

- All samples found in the wild install binaries into ~/Library/Preferences.
- ... WTF?
- Delivered via exploits, Flash, Word (etc?).
- Size is less than 1 megabyte.
- In latest versions the main backdoor module is packed with MPRESS.

You know what

I am MPRESS'ed!



Let’s hunt!

TrustedBSD

- Mandatory Access Control (MAC) Framework.
- Base technology used by Apple's sandbox.
- And iOS code signing.
- Hooks exist in many interesting places.
- We just need to install our callbacks.
- Very fast and stable, no hooking tricks required.

TrustedBSD

- The vnode_check_exec hook.
- Check if the subject can execute the passed vnode.
- Used by iOS code signing.
- Opportunity to observe what is going to be executed.
- Optionally, make a decision to execute or not.

TrustedBSD

```
typedef int mpo_vnode_check_exec_t(
 kauth_cred_t cred,
 struct vnode *vp,
 struct label *label,
 struct label *execlabel,/* NULLOK */
 struct componentname *cnp,
 u_int *csflags
);
```

TrustedBSD

- Use the vp parameter to retrieve the binary full path.
- vn_getpath() will construct the path to the vnode.
- If successful, match to /Users/xxxx/Library/Preferences.
- A clean system doesn't run Mach-O binaries from there.

TrustedBSD

```
char path[MAXPATHLEN] = {0};
int pathbuff_len = sizeof(path);
if ( vn_getpath(vp, path, &pathbuff_len) != 0 )
{
    ERROR_MSG("Can't build path to vnode!");
    return 0;
}
/* path will not be NULL here afterwards */
```

Kauth

- Kernel authorization framework.
- Introduced in Tiger.
- Refer to Technical Note TN2127.
- Can be used as a notification mechanism.
- The recommended interface for AV vendors.
- Four built-in scopes.

Kauth

- Process.
- Generic.
- File Operation (notification only).
- Vnode (the most powerful).

Kauth

- File operation scope is enough for today.
- Interested in KAUTH_FILEOP_CLOSE action.
- Notifies that a file system object is about to be closed.
- Idea is to detect Mach-O binaries written to Hacking Team's favorite path.

Kauth

- **Problem:** we get notified of all close operations.
- Too much noise.
- We just want those related to writes.
- There is a flag to save us!
- KAUTH_FILEOP_CLOSE_MODIFIED in arg2.
- Set if a <u>modified</u> file is being closed.

Kauth

```
static int
fileop_scope_listener(kauth_cred_t     credential,
                      void *           idata,
                      kauth_action_t   action,
                      uintptr_t        arg0, /* vnode ref */
                      uintptr_t        arg1, /* full path */
                      uintptr_t        arg2, /* flags */
                      uintptr_t        arg3)
```

Kauth

- Install a file operation listener.
- Ignore all actions except KAUTH_FILEOP_CLOSE.
- Only analyze close operations with flag KAUTH_FILEOP_CLOSE_MODIFIED set.
- Retrieve file info (size) from the vnode (arg0).
- Read file and verify if it's a (potentially valid) Mach-O.
- If yes, check path.

Kauth

```
/* retrieve the vnode attributes,
 * we can get a lot of vnode information from here */
struct vnode_attr vap = {0};
vfs_context_t context = vfs_context_create(NULL);
/* initialize the structure fields we are interested in
 * reference vn_stat_noauth() xnu/bsd/vfs/vfs_vnops.c
 */
VATTR_INIT(&vap);
VATTR_WANTED(&vap, va_mode);
VATTR_WANTED(&vap, va_type);
VATTR_WANTED(&vap, va_uid);
VATTR_WANTED(&vap, va_gid);
VATTR_WANTED(&vap, va_data_size);
VATTR_WANTED(&vap, va_flags);
int attr_ok = 1;
if ( vnode_getattr((vnode_t)arg0, &vap, context) != 0 )
{
  /* in case of error permissions and filesize will be bogus */
  ERROR_MSG("failed to vnode_getattr");
  attr_ok = 0;
}
/* release the context we created, else kabOOm! */
vfs_context_rele(context);
```

Conclusions

- A very simple kernel extension.
- Stable, fast, minimal impact in the system.
- Unless Hacking Team changes the folder it detects <u>every</u> single OS X infection.
- Compatible down to Snow Leopard.
- False positives issues?
- AV Monster PoC issues?

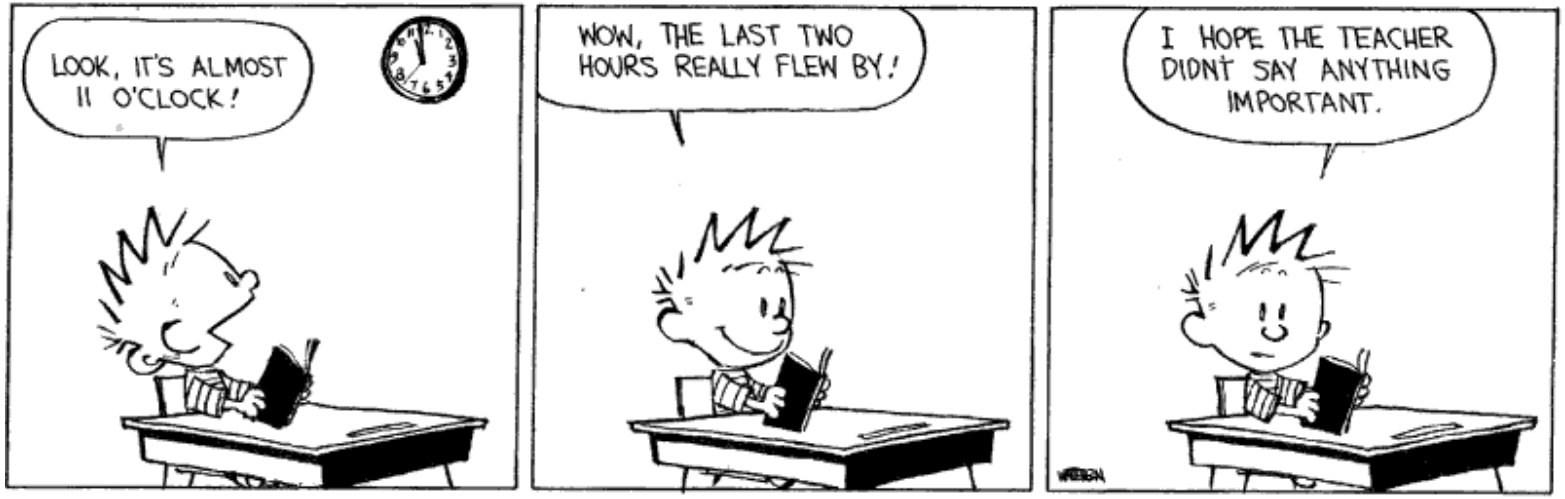
Conclusions

- Code signing and closing TrustedBSD issues.
- Apple doesn't give kext signing certificate to everyone.
- Must be afraid of kernel rootkits ;-).
- Or just wants to close access to the kernel except a selected few (I am a lucky one).
- Might release a binary for mass public consumption?


ASSUMPTIONS!

Greetings

You for spending time of your life listening to me, The Godfather aka Mr Thomas Lim for being great.



http://reverse.put.as

http://github.com/gdbinit

reverser@put.as

@osxreverser

#osxre @ irc.freenode.net