Sig•Mal

Meeting 8

What’s crack-a-lackin?

A gif with a bear wave in ascii art
Held this spot

RIP

Github Security

Keylogger

Security in Development

GITHUB

- Signing Commits
- Security Policy

KEYLOGGER

- Building in C++

Git and Github

Git ~ Version Control Software

Github ~ Cloud-based Project Management and Web Version Control

Version Control ~ Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions

GPG Keys

$ install gnupg (if not installed and on *nix System)

- Cryptography →
  - Cipher Text
  - `Secret messages`
  - Passwords (hopefully hashed)

For more info on signature commits and how to do them on Github

→ GPG → Validate / Verify that, that user is who they say they are.

CIA → Confidentiality, Integrity, Accessibility

GPG Keys

$ install gnupg (if not installed and on *nix System)

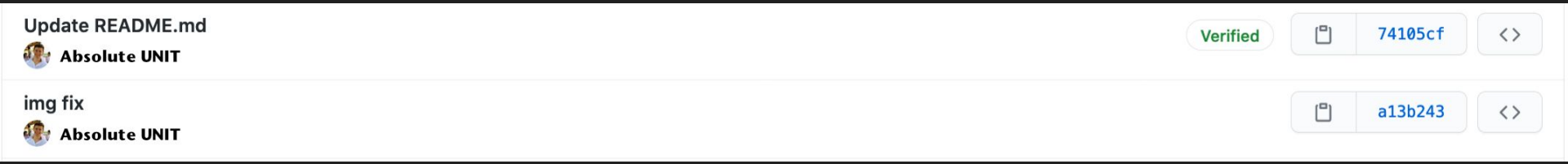
1. Generate Key
2. Share your Public Key with <thing> (not only for Github)
3. Get to signing :)

```
$ git commit -m “Message” -S

$ git commit -m “Message” --gpg-sign=<key-id>
```

Signing Commits (On Github)

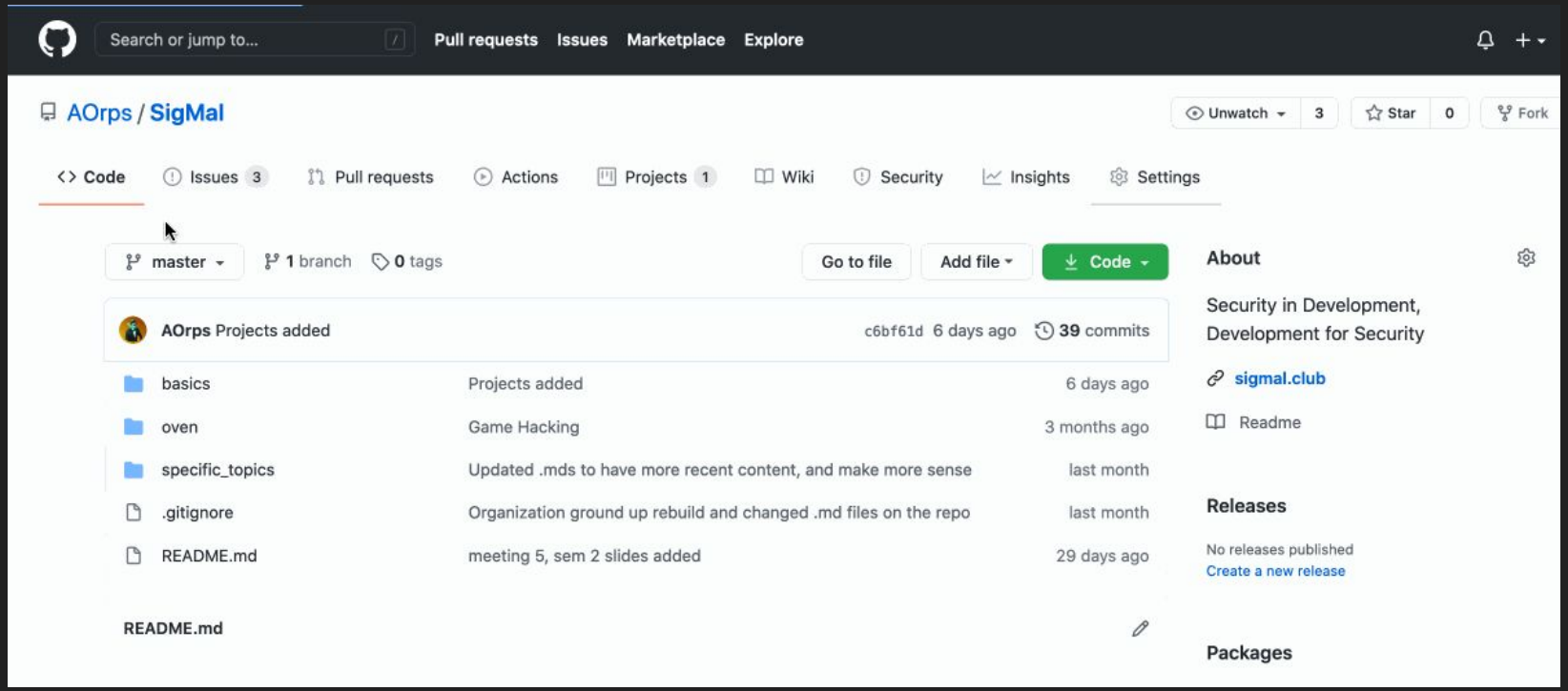
Ex:



+ Why is this important? Well you can validate that ‘Absolute UNIT’ is in fact ‘Absolute UNIT’.

→ If someone breaks into your account. They don’t do much damage to the system. If someone uses some ascii characters, it mitigates that too.

Github Security Policies



LIVE DEMO TIME

Keylogger

What is it?

It **logs** your **key**strokes.

How?

It will take some external library. Uses some function or class to get the “user input”. And sends it to some server or records it to a file. Normally, this is an executable ran in the background. Maybe ran as a service so not to alert the user.

Keylogger

- Let’s use the Operating System’s internal API / Library (idk what it is technically considered as)

- Internal OS API /* Building a Keylogger in c++ */
  - Windows → GetAsyncKeyState() → #include <winuser.h>
  - Linux → #include <linux/input.h>
  - Mac → #include <ApplicationServices/ApplicationServices.h>

Keylogger

- Group up and Determine for which OS you want the Keylogger to target. Copy the program, try to figure out what it does. And see if you can add onto it. If you were an attacker, how would you put your Keylogger.

+ MacOS → Click Me
+ Linux → Click Me
+ Windows → Click Me

Ideas to add: Regex and Parsing/Forming Data, Persistence, Delivery of the Keylogger

Sources

- Git and Github
- Managing Commit Signature Verification
- MacOS Keylogger Example
- Linux Keylogger Example
- Windows Keylogger Example