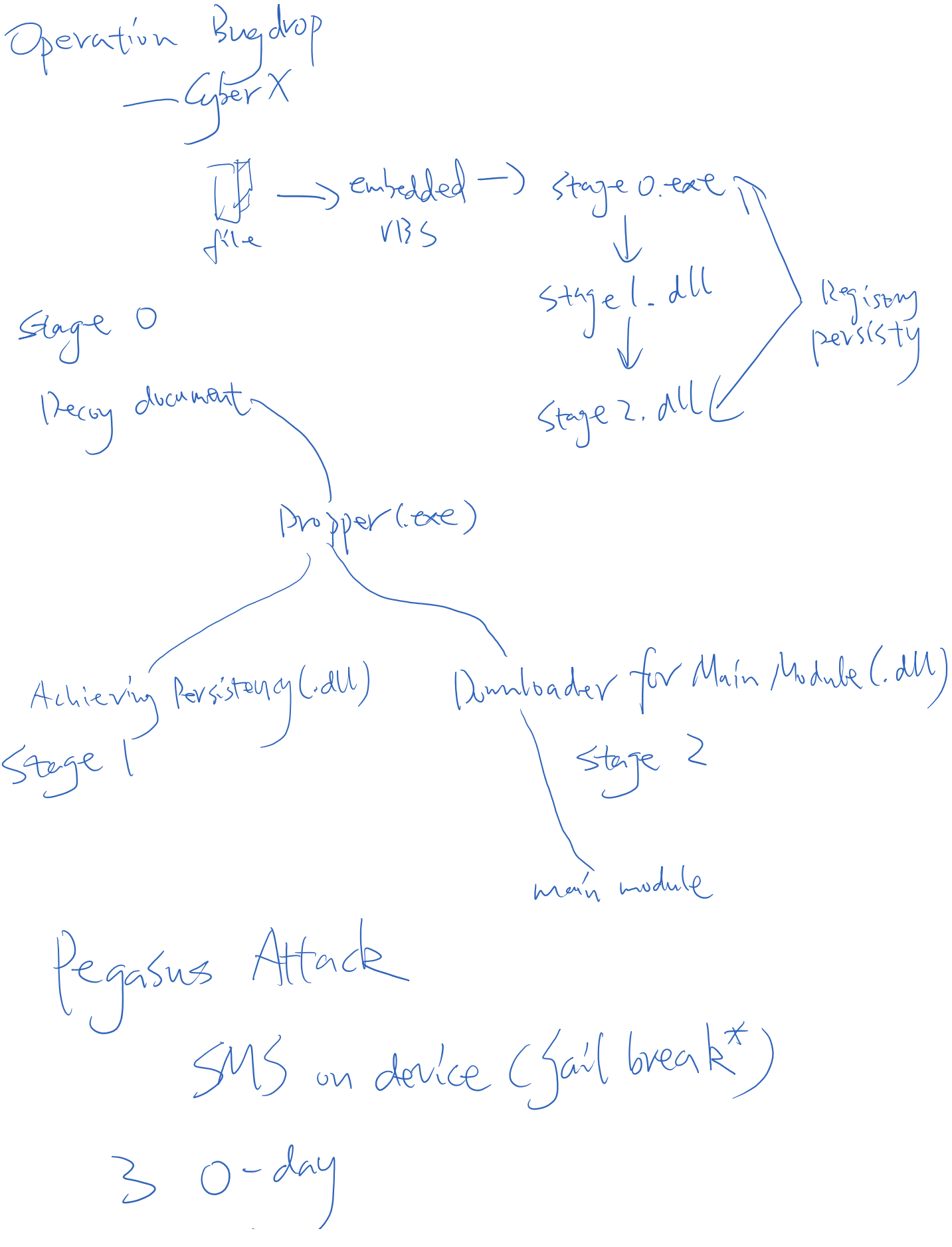
Mar 23rd

Thursday, March 23, 2017 1:54 PM



5 0-day

iOS Kernel Privileges

1. webKit Memory Corruption
   run on iphone
2. Kernel Information Leak
   attack iOS kernel
3. Kernel Use-After-free
   can execute code
   Has kernel location
   full access

Prevention

update iOS
antivirus software

Web Hacking II

cookies, SOP, XSS and CSRF

Setting Cookies

- Key = value pair
- AKA Cookie Crumb

Expiration Date

- If not set cookie is Session cookie

Domain Allowed to access cookie

For Domain=.foo.com:

Who can access?

- Bar.foo.com
- #.com
- Foo.com/bar

If .foo.com different from foo.com?

Yes

Cookies are key=value pairs with metadata

- HostOnly
  - Can only be read by same domain that set it, foo.example.com sets it so bar.example.com cannot read it if HostOnly=1
- Session
  - A session cookie expires when the user navigates away from the website
- Secure
  - Cookie can only be sent over https
- HttpOnly
  - Cannot be access with JavaScript

Zombie Cookies

A zombie cookie is a cookie that is automatically recreated after being deleted. This is accomplished by storing the cookie's content in multiple locations

In the News

- EU requires users agree it have users cited on their pages
- Confining the Power of JavaScript Scripts
  - Given all power
- Same Origin policy

  Browsers provide isolation for JS scripts via the same origin Policy (SOP)
- Same Origin Policy
  - Are subdomains included in SOP policy?

ZAA: Subverting the same origin policy

It'd be Bad if an attacker from evil.com can fool your browser into executing script of their choice… with your browser believing the script's origin to be some other site, like bank.com

Your browser is none the wiser, and executes it within the same origin as the bank.com server

Two types of XSS

In a stored XSS attack, the attacker leaves their script lying around on bank.com server

…and the server later unwittingly sends it to your browser

Protecting Servers Against XSS

OWASP = Open Web Application Security Project

The best way to protect against XSS attacks:

Ensure that your app validates all headers, cookie, query string, from fields, and hidden fields against a rigorous specification of what should be allowed
Do not attempt to identify active content and remove, filter, or sanitize it.