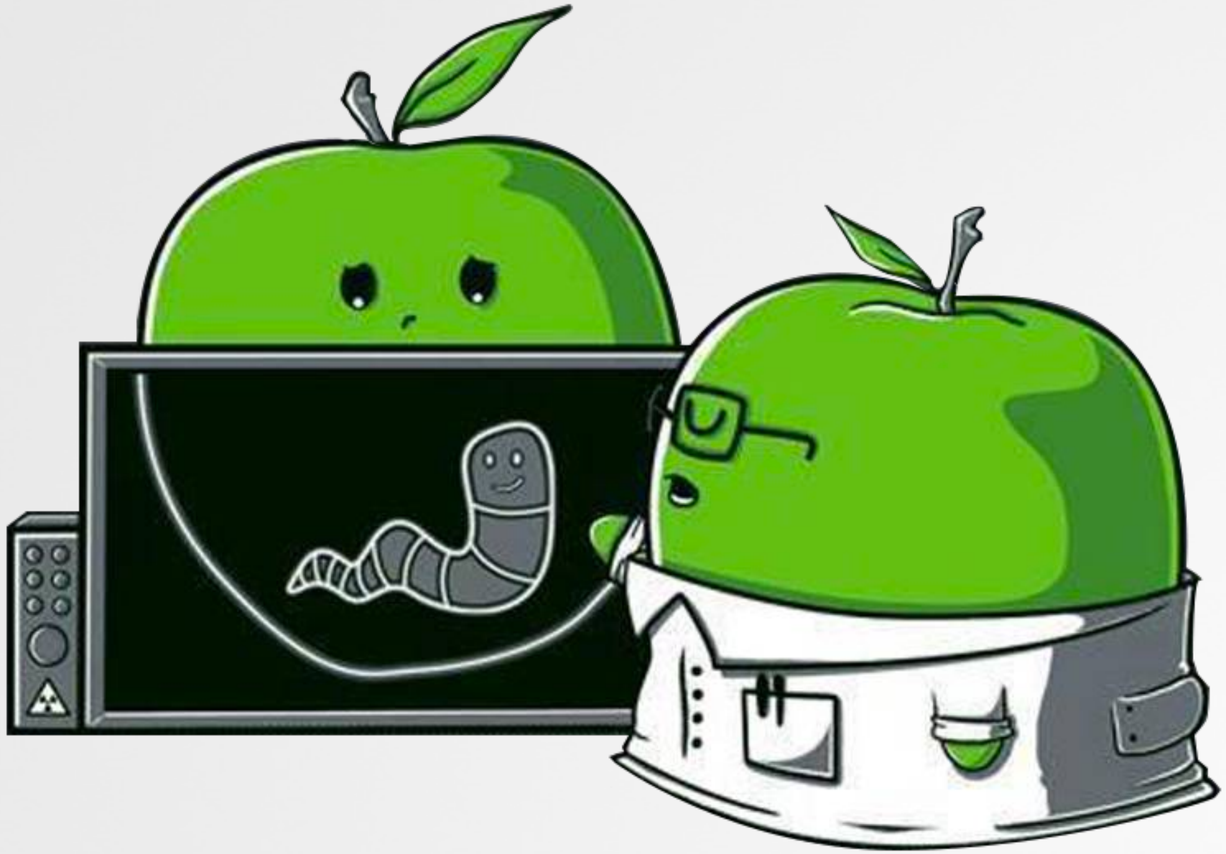
Did Apple Solve Persistence?

Demystifying macOS's Background Task Management



Bios

Christopher Lopez
(Kandji)

Patrick Wardle
(Objective-See Foundation)

What You Will Learn

Although the talk is largely focused on macOS's Background Task Management (BTM) - we'll also cover topics of reversing, malware detection, macOS internals, and more!

1 

Internals of macOS's BTM

2 

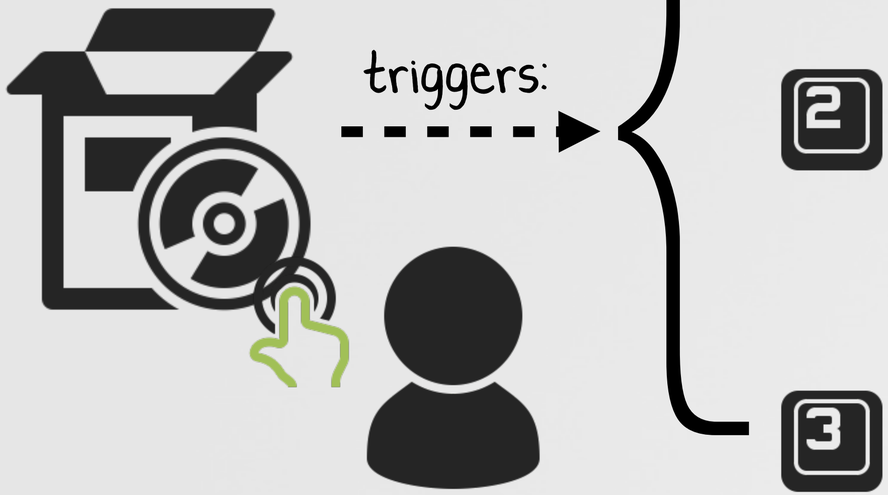
Leveraging BTM to detect malware

3 

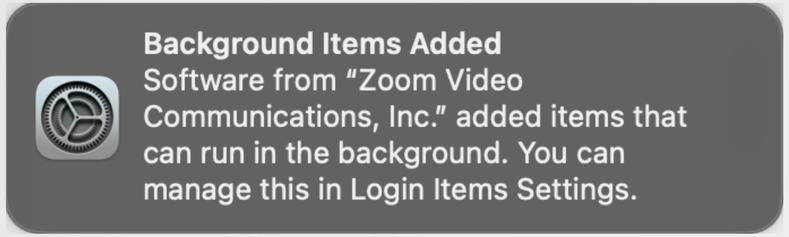
Bypassing BTM (alerts & ES messaging)

So What is Background Task Management (BTM)?

governance of persistent items ("background tasks")



1



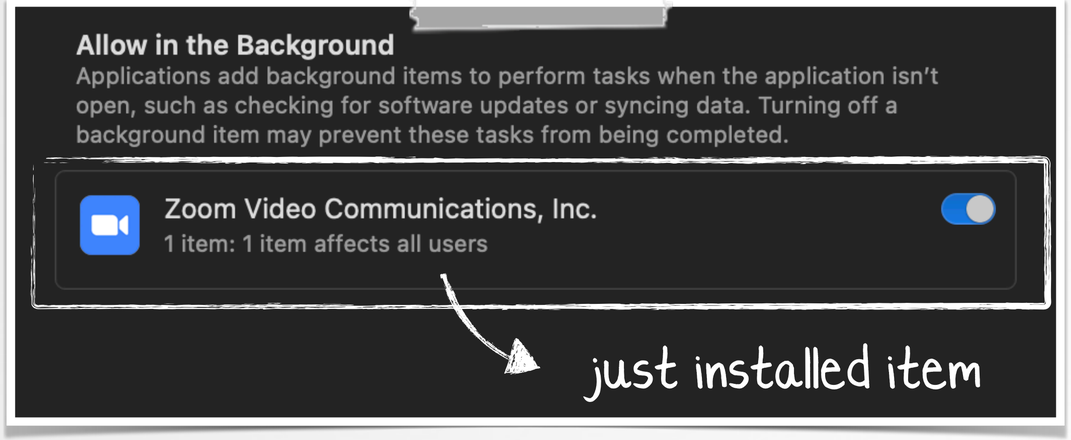
System Alert shown to user

2

```
# ./btm_monitor
New Event: 'ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD'

Item: /Library/PrivilegedHelperTools/us.zoom.ZoomDaemon
```

Endpoint Security event

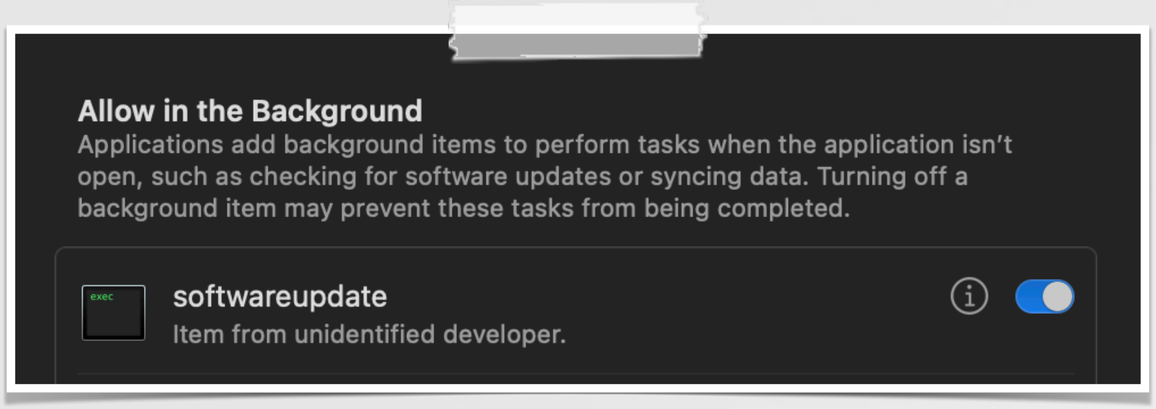
3

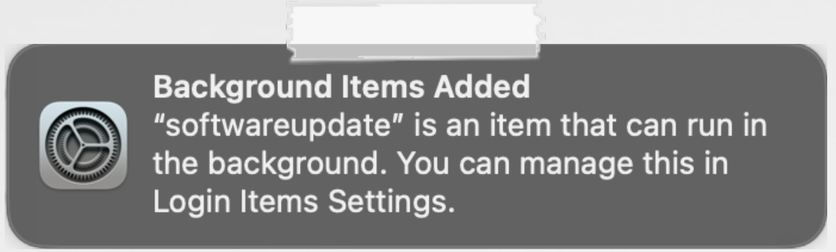


System "registration"

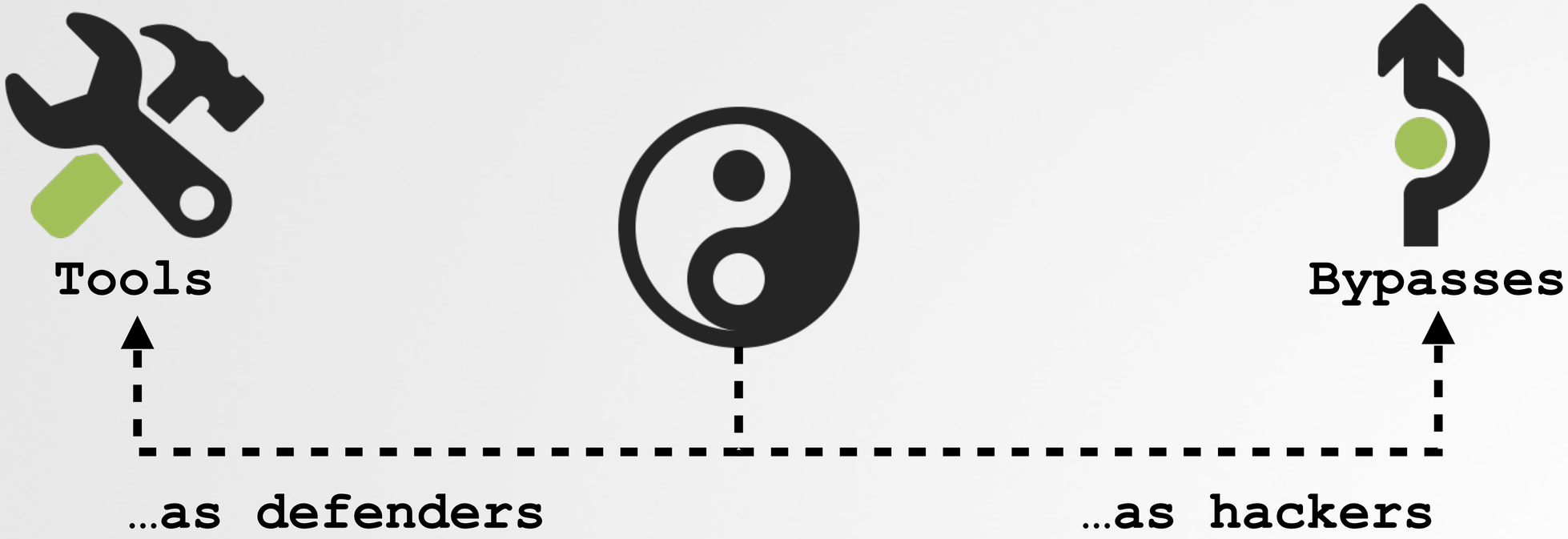
Who Cares?

well, we do!

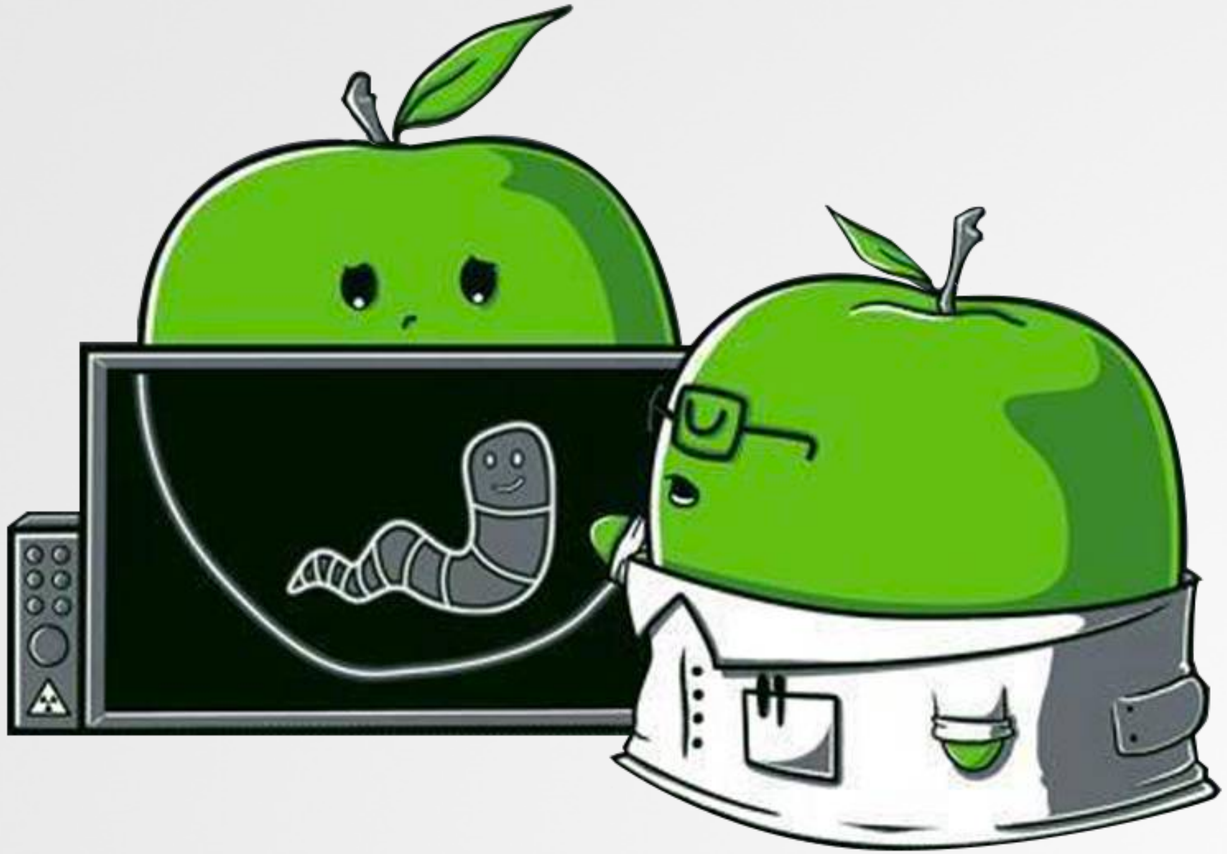





```
# ./btm_monitor
'ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD'
~/Library/LaunchAgents/com.apple.softwareupdate.plist
```



BTM Internals



Background Items

Daemon

1. /System/Library/PrivateFrameworks/BackgroundTaskManagement.framework/Versions/A/Resources/backgroundtaskmanagementd

Agent

2. /System/Library/PrivateFrameworks/BackgroundTaskManagement.framework/Support/BackgroundTaskManagementAgent.app

BACKGROUNDTASKMANAGEMENTD

daemon for managing background items

```
% log stream ...

2023-03-20 10:52:28.948037-0400 0x9268  Default  0x1ee83  242 0

backgroundtaskmanagementd: [com.apple.backgroundtaskmanagement:main] registerLaunchItem:
updated item: uuid=388DB692-E9E3-4A79-9737-F1F192978ECE, name=steamclean, type=legacy
agent, disposition=[enabled, allowed, visible, not notified],
identifier=com.valvesoftware.steamclean, url=file:///Users/l0psec/Library/LaunchAgents/
com.valvesoftware.steamclean.plist
```

example unified log event

BACKGROUNDTASKMANAGEMENTD

-[BTMItem createItemRecord]

```
000000010000fc5c        ldr     x0, =__objc_class_ItemRecord_class      ;
argument "class" for method imp___auth_stubs__objc_alloc,
__objc_class_ItemRecord_class
000000010000fc60        bl      imp___auth_stubs__objc_alloc            ;
objc_alloc
000000010000fc64        mov     x19, x0
000000010000fc68        mov     x0, x20
000000010000fc6c        bl      type                                    ; type
000000010000fc70        mov     x21, x0
000000010000fc74        mov     x0, x20
000000010000fc78        bl      identifier                              ; identifier

000000010000fc80        bl
imp___auth_stubs__objc_retainAutoreleasedReturnValue ;
objc_retainAutoreleasedReturnValue
000000010000fc84        mov     x22, x0
000000010000fc88        mov     x0, x19
000000010000fc8c        mov     x2, x21
000000010000fc90        mov     x3, x22
000000010000fc94        bl      initWithType:identifier:                ;
initWithType:identifier:
000000010000fc98        mov     x19, x0
000000010000fc9c        mov     x0, x22                                 ; argument
"instance" for method imp___auth_stubs__objc_release
```

```
000000010000fca0        bl      imp___auth_stubs__objc_release          ;
objc_release
000000010000fca4        mov     x0, x20
000000010000fca8        bl      uuid                                    ; uuid
000000010000fcac        mov     fp, fp
000000010000fcb0        bl
imp___auth_stubs__objc_retainAutoreleasedReturnValue ;
objc_retainAutoreleasedReturnValue
000000010000fcb4        mov     x21, x0
000000010000fcb8        mov     x0, x19
000000010000fcbc        mov     x2, x21
000000010000fcc0        bl      setUuid:                                ; setUuid:
000000010000fcc4        mov     x0, x21                                 ; argument
"instance" for method imp___auth_stubs__objc_release
000000010000fcc8        bl      imp___auth_stubs__objc_release          ;
objc_release
000000010000fccc        mov     x0, x20
000000010000fcd0        bl      url                                     ; url
000000010000fcd4        mov     fp, fp
000000010000fcd8        bl
imp___auth_stubs__objc_retainAutoreleasedReturnValue ;
objc_retainAutoreleasedReturnValue
000000010000fcdc        mov     x21, x0
000000010000fce0        mov     x0, x19
000000010000fce4        mov     x2, x21
000000010000fce8        bl      setUrl:                                 ; setUrl:
```

Item Records class

Properties

```
@class ItemRecord : NSObject<NSCopying, NSSecureCoding> {
                    ;   @property uuid
                    ;   @property identifier
                    ;   @property disposition
                    ;   @property enabled
                    ;   @property allowed
                    ;   @property generation
                    ;   @property url
                    ;   @property modificationDate
                    ;   @property executablePath
                    ;   @property executableModificationDate
                    ;   @property name
                    ;   @property developerName
                    ;   @property isUnknownDeveloperRecord
                    ;   @property teamIdentifier
                    ;   @property lightweightRequirement
                    ;   @property type
                    ;   @property isDaemon
                    ;   @property managed
                    ;   @property bookmark
                    ;   @property bundleIdentifier
                    ;   @property associatedBundleIdentifiers
                    ;   @property container
                    ;   @property embeddedItems
                    ;   @property dumpVerboseDescription
```

How Does backgroundtaskmanagementd know?

-[FSEventsObserver initWithURL:filterPath:]
-[FSEventsObserver handleEventWithID:url:eventFlags:]

```
@protocol FSEventsObserverDelegate<NSObject>

@optional
- (void)eventObserver:(id)v1 didAddFileWithURL:(id)v2;
- (void)eventObserver:(id)v1 didRemoveFileWithURL:(id)v2;
- (void)eventObserver:(id)v1 didRenameFileWithURL:(id)v2;
- (void)eventObserver:(id)v1 didChangeFileWithURL:(id)v2;
- (void)eventObserver:(id)v1 didRemoveRootWithURL:(id)v2;
@end
```

```
int FSEventStreamCreate(int allocator, int callback, void * context, int pathsToWatch, int sinceWhen, int latency, int flags) {
    r0 = _FSEventStreamCreate_10002c048(allocator, callback, context, pathsToWatch, sinceWhen, latency, flags);
    return r0;
}
```

```
000000010001e840        mov     x0, x20
000000010001e844        bl      delegate                        ; delegate
000000010001e848        mov     fp, fp
000000010001e84c        bl      imp___auth_stubs__objc_retainAutoreleasedReturnValue ; objc_retainAutoreleasedReturnValue
000000010001e850        mov     x23, x0
000000010001e854        mov     x2, x20
000000010001e858        mov     x3, x19
000000010001e85c        bl      eventObserver:didAddFileWithURL:        ; eventObserver:didAddFileWithURL:
000000010001e860        mov     x0, x23                         ; argument "instance" for method imp___auth_stubs__objc_release
000000010001e864        bl      imp___auth_stubs__objc_release          ; objc_release
```

Where are Items stored?

+[BTMStore storeNameForDatabaseVersion:]

```
0000000100017368        pacibsp                                 ; Objective C Implementation defined at 0x1000246dc (class method), DATA XREF=0x1000246dc
000000010001736c        sub     sp, sp, #0x20
0000000100017370        stp     fp, lr, [sp, #0x10]
0000000100017374        add     fp, sp, #0x10
0000000100017378        nop
000000010001737c        ldr     x0, =_OBJC_CLASS_$_NSString             ; _OBJC_CLASS_$_NSString
0000000100017380        str     x2, [sp, #0x10 + var_10]
0000000100017384        adr     x2, #0x10002dd38                        ; @"BackgroundItems-v%ld.btm"
0000000100017388        nop
000000010001738c        bl      stringWithFormat:                       ; stringWithFormat:
0000000100017390        ldp     fp, lr, [sp, #0x10]
0000000100017394        add     sp, sp, #0x20
0000000100017398        retab
```

BTM version number changes for different OS versions

XPC COMMUNICATIONS

+[* listener:shouldAcceptNewConnection:]

-[**SharedFileListService** listener:shouldAcceptNewConnection:]
-[**BTMService** listener:shouldAcceptNewConnection:]
-[**DaemonNotificationManager** listener:shouldAcceptNewConnection:]

```
loc_1000047a4:
    [interfaceWithProtocol @protocol(V2ServiceInterface)() retain];
    setExportedInterface:();
    [r25 release];
    setExportedObject:();
    setInterruptionHandler:();
    setInvalidationHandler:();
    resume();
    addObject:();
    [r24 release];
    [r23 release];
    [r22 release];
    r21 = 0x1;
    goto loc_100004864;
```

```
@protocol V2ServiceInterface {
                    ;   -fetchUserSettingsForUID:reply:
                    ;   -saveUserSettings:uid:reply:
                    ;   -launchdWillScanPath:reply:
                    ;   -launchdDidScanPath:reply:
                    ;   -registerLaunchItemWithAuditToken:parentURL:type:relativeURL:configuration:uid:reply:
                    ;   -unregisterLaunchItemWithAuditToken:parentURL:type:relativeURL:uid:reply:
                    ;   -invalidateLaunchItemWithBundleURL:type:itemURL:configuration:uid:reply:
                    ;   -canLaunchItemWithAppURL:type:relativeURL:configuration:uid:reply:
                    ;   -itemDispositionWithAppURL:type:relativeURL:configuration:uid:reply:
                    ;   -effectiveItemDispositionWithAppURL:type:relativeURL:configuration:uid:reply:
                    ;   -getItemWithIdentifier:uid:reply:
                    ;   -getItemsMatching:uid:reply:
                    ;   -getItemWithUUID:uid:reply:
                    ;   -itemWithAppURL:type:relativeURL:configuration:uid:reply:
                    ;   -addItem:uid:reply:
                    ;   -updateItem:uid:reply:
                    ;   -updateItems:uid:reply:
                    ;   -removeItemWithIdentifier:uid:reply:
                    ;   -removeItemWithUUID:uid:reply:
                    ;   -mdmPayloadForIdentifier:reply:
                    ;   -setMDMPayload:identifier:updateImmediately:reply:
                    ;   -fetchUserIdentifiersAndUIDsWithReply:
                    ;   -dumpDatabaseWithAuthorization:reply:
                    ;   -eraseDatabaseWithAuthorization:reply:
                    ;   -setAlwaysPostNotifications:reply:
                    ;   -getAlwaysPostNotificationsWithReply:
```

Entitlement Check Example:

```
00000001000046fc        adr     x2, #0x100030e10                ;
@"com.apple.private.backgroundtaskmanagement.manage", CODE XREF=-[BTMService
listener:shouldAcceptNewConnection:]+304
0000000100004700        nop
0000000100004704        mov     x0, x20
0000000100004708        bl      valueForEntitlement:            ; valueForEntitlement:
```

WHAT DO THE RECORDS LOOK LIKE?

Example using dumpBTM :)

```
#26
  UUID:              2CA629F0-B6A3-40C5-AABD-0DAF91B98BD5
  Name:              steamclean
  Developer Name:    Steam
  Team Identifier:   MXGJJ98X76
  Type:              curated legacy agent  (0x90008)
  Disposition:       [enabled allowed visible notified] (11)
  Indentifier:       com.valvesoftware.steamclean
  URL:               file:///Users/<username>/Library/LaunchAgents/com.valvesoftware.steamclean.plist
  Executable Path:   /Users/<username>/Library/Application Support/Steam/SteamApps/steamclean
  Generation:        6
  Assoc. Bundle IDs: [com.valvesoftware.steam]
  Parent Identifier: Steam
```

curated legacy agent?

WHAT IS CURATED?

"Approved apps???"

Type: curated developer (0x80020)
Type: curated legacy daemon (0x90010)
Type: managed curated developer (0xa0020)
Type: managed curated legacy agent (0xb0008)
Type: managed curated legacy daemon (0xb0010)

Different curated examples

sub_10000d5fc

```
000000010000d620        ldr       x0, [x20, #0x30]                  ;
objc_cls_ref_NSBundle,_OBJC_CLASS_$_NSBundle
000000010000d624        adr       x2, #0x100031398                    ;
@"com.apple.coreservices.BackgroundTaskManagement"
000000010000d628        nop
000000010000d62c        bl        bundleWithIdentifier:               ; bundleWithIdentifier:

000000010000d634        bl        imp___auth_stubs__objc_retainAutoreleasedReturnValue ;
objc_retainAutoreleasedReturnValue
000000010000d638        mov       x21, x0
000000010000d63c        adr       x2, #0x1000313b8                    ; @"attributions"
000000010000d640        nop
000000010000d644        adr       x3, #0x100031378                    ; @"plist"
000000010000d648        nop
000000010000d64c        bl        pathForResource:ofType:             ; pathForResource:ofType:

000000010000d65c        adrp      x22, #0x100037000
000000010000d660        ldr       x0, [x22, #0xf60]                 ;
objc_cls_ref_NSDictionary,_OBJC_CLASS_$_NSDictionary
000000010000d664        mov       x2, x19
000000010000d668        bl        dictionaryWithContentsOfFile:       ; dictionaryWithContentsOfFile:
```

attributions.plist in Resources directory

ATTRIBUTIONS.PLIST

/System/Library/PrivateFrameworks/BackgroundTaskManagement.framework/Resources/attributions.plist

```
"Adobe_Genuine_Software_Integrity_Service" => {
    "AssociatedBundleIdentifiers" => [
      0 => "com.adobe.acc.AdobeCreativeCloud"
    ]
    "Attribution" => "Adobe Creative Cloud"
    "ProgramArguments" => [
      0 => "/Library/Application Support/Adobe/AdobeGCClient/
AGSService"
    ]
    "TeamIdentifier" => "JQ525L2MZD"
```

example match

Apple does mention this plist in their docs:

https://support.apple.com/guide/deployment/manage-login-items-background-tasks-mac-depdca572563/web

What about Managed Items?

MDM? Profiles?

```
2023-09-30 07:56:27.109826-0400 0x548db0   Default     0x0                   24882  0
BTMProfileService: [com.apple.backgroundtaskmanagement:mcx] cpInstallPayload: called for
A3DE68A0-4FC4-4F70-82E3-81A49485FDE7

2023-09-30 07:56:30.110404-0400 0x547f19   Default     0xf2377e              15243  0
backgroundtaskmanagementd: [com.apple.backgroundtaskmanagement:mcx] Rule 'Label: com.c.test'
from profile payload A3DE68A0-4FC4-4F70-82E3-81A49485FDE7 matched item 888EDEEE-1A43-4DED-
A637-5B6A62E8C3DB
```

Type: managed app (0x20002)
Type: managed curated developer (0xa0020)
Type: managed curated legacy agent (0xb0008)
Type: managed curated legacy daemon (0xb0010)
Type: managed developer (0x20020)
Type: managed legacy agent (0x30008)
Type: managed legacy daemon (0x30010)
Type: managed login item (0x20004)

Profile Example

Installed via MDM

```
2023-09-30 07:56:27.217119-0400 0x548d71   Default     0x0
24880   0    mdmclient: (ConfigurationProfiles)
[com.apple.ManagedClient:ManagedClient] ESNotifyProfile
(Add): ID: com.kandji.profile.custom.40facf4d-521f-4dde-
bcd7-726bf1ab5f76  UUID: 40facf4d-521f-4dde-
bcd7-726bf1ab5f76  Org: com.test.btm  Name: BTM_Test
Scope: system  Source: 0

2023-09-30 07:56:27.217281-0400 0x548d71   Default     0x0
24880   0    mdmclient: [com.apple.ManagedClient:MDMDaemon]
[0:MDMDaemon:<0x548d71>] Installed configuration profile:
BTM_Test (com.kandji.profile.custom.40facf4d-521f-4dde-
bcd7-726bf1ab5f76:40facf4d-521f-4dde-bcd7-726bf1ab5f76) for
<Computer> (Source: MDM)

2023-09-30 07:56:27.351083-0400 0x42ec45   Default     0x0
5393   21   AssetCache: [com.apple.AssetCache:builtin]
Notification user info: {
    ProfileAction = Install;
    ProfileTypes =     (
        "com.apple.servicemanagement"
    );
    ProfileUUID = "40facf4d-521f-4dde-bcd7-726bf1ab5f76";
    ProfileUserUID = 501;
    ProfileUsername = l0psec;
}
```

profiles show -type configuration

_computerlevel[2] attribute: name: **BTM_Test**
_computerlevel[2] attribute: configurationDescription: (null)
_computerlevel[2] attribute: installationDate: 2023-09-30 11:56:27 +0000
_computerlevel[2] attribute: organization: com.test.btm
_computerlevel[2] attribute: profileIdentifier: com.kandji.profile.custom.40facf4d-521f-4dde-bcd7-726bf1ab5f76
_computerlevel[2] attribute: profileUUID: **40facf4d-521f-4dde-bcd7-726bf1ab5f76**
_computerlevel[2] attribute: profileType: Configuration
_computerlevel[2] attribute: removalDisallowed: FALSE
_computerlevel[2] attribute: version: 1
_computerlevel[2] attribute: containsComputerItems: TRUE
_computerlevel[2] attribute: installedByMDM: TRUE
_computerlevel[2] attribute: internaldata: TRUE
_computerlevel[2] payload count = 1
_computerlevel[2] payload[1] name = Service Management - Managed Login Items
_computerlevel[2] payload[1] description = (null)
_computerlevel[2] payload[1] type = com.apple.servicemanagement
_computerlevel[2] payload[1] organization = (null)
_computerlevel[2] payload[1] identifier = com.apple.servicemanagement.A3DE68A0-4FC4-4F70-82E3-81A49485FDE7
_computerlevel[2] payload[1] uuid = **A3DE68A0-4FC4-4F70-82E3-81A49485FDE7**

profileUUID = **40facf4d-521f-4dde-bcd7-726bf1ab5f76**

payload[1] uuid = **A3DE68A0-4FC4-4F70-82E3-81A49485FDE7**

BTMProfileService

Rules? Matched Items?

Configuration Profile Installation via MDM

BTMProfileService (via XPC)

-[BTMProfileService cpInstallPayload:profile:existingProfile:actions:error:]:

-(int)cpInstallPayload:(int)arg2 profile:(int)arg3 existingProfile:(int)arg4 actions:(int)arg5 error:(int)arg6 {

backgroundtaskmanagementd method

-[BTMStore checkMDMRulesForRecord:]:

Rule 'Label: com.c.test' from profile payload A3DE68A0-4FC4-4F70-82E3-81A49485FDE7 matched item 888EDEEE-1A43-4DED-A637-5B6A62E8C3DB

Rules

BundleIdentifier
BundleIdentifierPrefix
Label
LabelPrefix
TeamIdentifier

BackgroundItems-v%ld.btm

Good opportunities for further research :)

BackgroundTaskManagementAgent

agent in charge of notifications

```
2023-03-20 10:52:28.949931-0400 0x4f4e     Default     0x1ee83                564    0
BackgroundTaskManagementAgent: [com.apple.backgroundtaskmanagement:main] Posting advisory
notification request: identifier=388DB692-E9E3-4A79-9737-F1F192978ECE, body='“steamclean” is
an item that can run in the b…'

2023-03-20 10:52:28.949960-0400 0x4f4e     Default     0x1ee83                564    2
BackgroundTaskManagementAgent: (UserNotifications) [com.apple.UserNotifications:Connections]
[com.apple.BTMNotificationAgent] Adding notification request 928B-45D8 to destinations:
Default

2023-03-20 10:52:28.950641-0400 0x4f4e     Default     0x1ee83                564    2
BackgroundTaskManagementAgent: (UserNotifications) [com.apple.UserNotifications:Connections]
[com.apple.BTMNotificationAgent] Added notification request: [ hasError: 0
hasCompletionHandler: 1 ]
```

How do notifications work?

UserNotificiation Bundles

-[NotificationManager postManagedItemNotificationForItem:container:]

```
0000000100015180        mov     x0, x21                         ; CODE XREF=-[NotificationManager postManagedItemNotificationForItem:container:]+160
0000000100015184        mov     w2, #0x1
0000000100015188        bl      setHaveManagedItemNotification:         ; setHaveManagedItemNotification:

0000000100015190        ldr     x0, =_OBJC_CLASS_$_UNMutableNotificationContent ; _OBJC_CLASS_$_UNMutableNotificationContent
0000000100015194        bl      imp___auth_stubs__objc_opt_new          ; objc_opt_new
0000000100015198        mov     x22, x0
000000010001519c        adr     x0, #0x100025618                ; @"NOTIFICATION_TITLE_MANAGED_ITEM"
00000001000151a0        nop
00000001000151a4        bl      sub_100013b50                   ; sub_100013b50
```

sub_100013b50

```
0000000100013b6c        ldr     x19, =_OBJC_CLASS_$_NSBundle ; _OBJC_CLASS_$_NSBundle
0000000100013b70        bl      imp___auth_stubs__objc_retain           ; objc_retain
0000000100013b74        mov     x20, x0
0000000100013b78        adr     x2, #0x1000256d8                ; @"/System/Library/UserNotifications/Bundles/com.apple.BTMNotificationAgent.bundle"
0000000100013b7c        nop
0000000100013b80        mov     x0, x19
0000000100013b84        bl      bundleWithPath:                 ; bundleWithPath:

0000000100013b8c        bl      imp___auth_stubs__objc_retainAutoreleasedReturnValue ; objc_retainAutoreleasedReturnValue
0000000100013b90        mov     x21, x0
0000000100013b94        adr     x3, #0x1000256f8                ; @""
0000000100013b98        nop
0000000100013b9c        mov     x2, x20
0000000100013ba0        mov     x4, #0x0
0000000100013ba4        bl      localizedStringForKey:value:table:      ; localizedStringForKey:value:table:
```

User Notification Bundles?

BackgroundTaskManagementAgent Entitlements

Entitlements -
- [Key] com.apple.private.CoreAuthentication.SPI
- [Value]
  - [Bool] true
- [Key] com.apple.private.LocalAuthentication.CallerName
- [Value]
  - [Bool] true
- [Key] com.apple.private.backgroundtaskmanagement.manage
- [Value]
  - [Bool] true
- [Key] com.apple.private.backgroundtaskmanagement.responses
- [Value]
  - [Bool] true
- [Key] com.apple.private.tcc.allow
- [Value]
  - [Array]
    - [String] kTCCServiceSystemPolicyAllFiles
- [Key] com.apple.private.coreservices.canaccessanysharedfilelist
- [Value]
  - [String] read-write
- [Key] com.apple.private.sharedfilelist.AllowRestrictedItemPropertyUpdates
- [Value]
  - [Bool] true
- **[Key] com.apple.private.usernotifications.bundle-identifiers**
- **[Value]**
  - **[Array]**
    - **[String] com.apple.BTMNotificationAgent**

/System/Library/UserNotifications/Bundles/
com.apple.BTMNotificationAgent.bundle/Contents/Resources/Localizable.loctable

"en" => {

"APPROVAL_AUTH_REASON" => "You must authenticate as an administrator to allow background items for all users."

"APPROVE" => "Allow"

"DONT_APPROVE" => "Don’t Allow"

"NOTIFICATION_BODY_APP_LOGIN_ITEM_ADVISORY" => "“%@” will open automatically when you log in. You can manage this in Login Items Settings."

"NOTIFICATION_BODY_APPROVAL" => "“%@” added items that can run in the background for all users. Do you want to allow this?"

"NOTIFICATION_BODY_BACKGROUND_ITEM_ADVISORY" => "“%@” added items that can run in the background. You can manage this in Login Items Settings."

"NOTIFICATION_BODY_BACKGROUND_ITEM_ADVISORY_DEVELOPER" => "Software from “%@” added items that can run in the background. You can manage this in Login Items Settings."

"NOTIFICATION_BODY_BACKGROUND_ITEM_ADVISORY_UNKNOWN" => "“%@” is an item that can run in the background. You can manage this in Login Items Settings."

"NOTIFICATION_BODY_MANAGED_ITEM" => "Your organization added items that can run in the background. You can view these in Login Items Settings."

"NOTIFICATION_TITLE_APP_LOGIN_ITEM_ADVISORY" => "Login Item Added"

"NOTIFICATION_TITLE_APPROVAL" => "Background Items Added"

"NOTIFICATION_TITLE_BACKGROUND_ITEM_ADVISORY" => "Background Items Added"

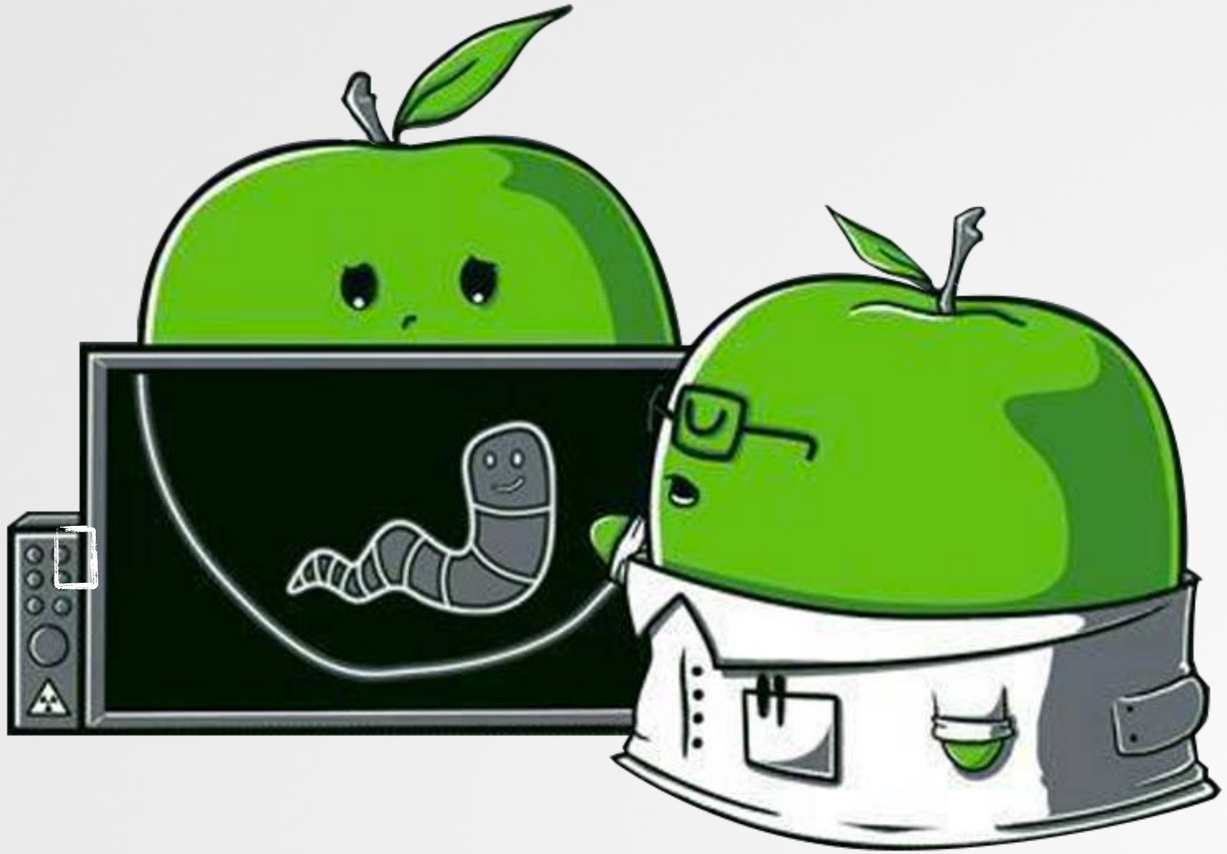
"NOTIFICATION_TITLE_MANAGED_ITEM" => "Managed Login Items Added"

"SNOOZE_ONE_DAY" => "Snooze for 1 day"

"SNOOZE_ONE_WEEK" => "Snooze for 1 week"

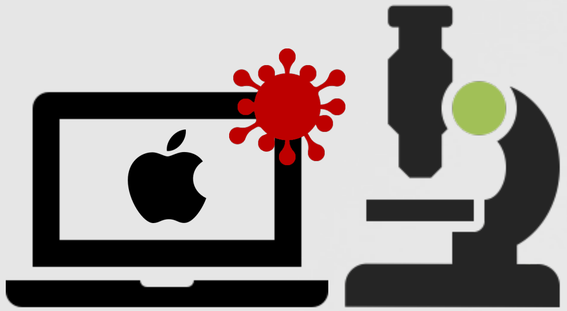
Tools (Part I)

(reactively) detecting persistent malware via BTM



Mac Malware

...the vast majority persists!



The Mac Malware of 2022 👾

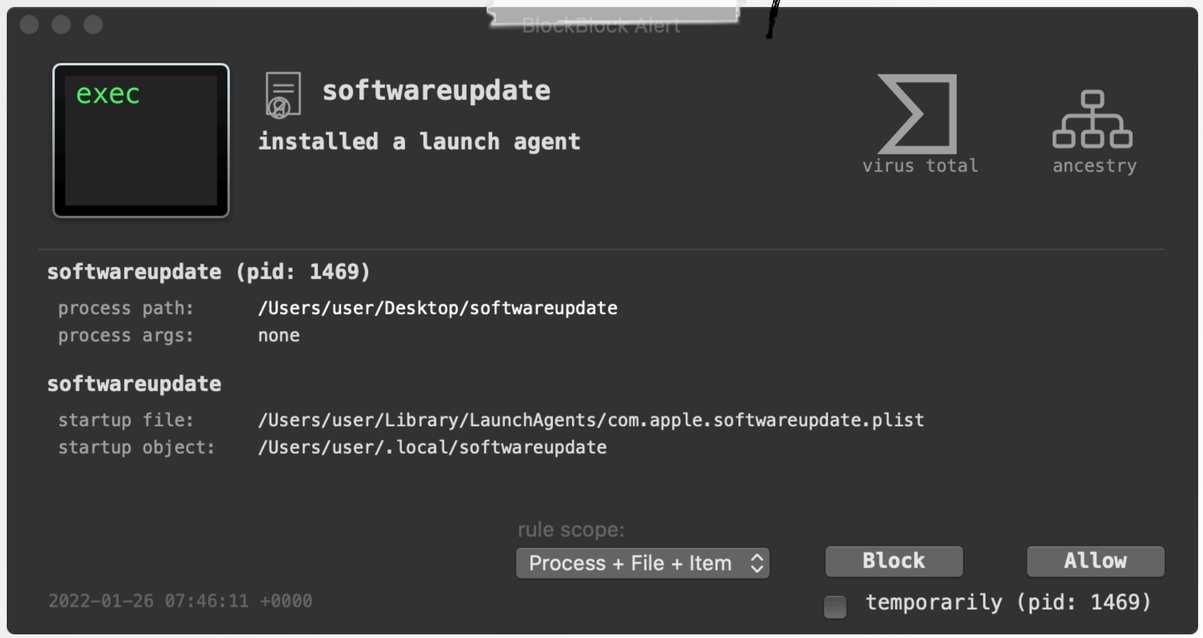
A comprehensive analysis of the year's new malware

by: Patrick Wardle / January 1, 2023

2022: ~80% persisted
(all as launch items)

2021: ~90% persisted
(all as launch items)

alert user when something persists !



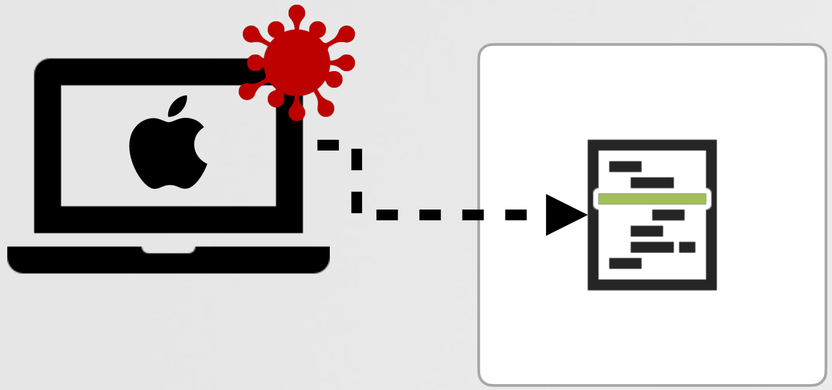
BlockBlock
...block block'ing since 2015!

OSX.DazzleSpy

…persistently deployed via Safari 0days

```
% strings - DazzleSpy/softwareupdate
...
%@/Library/LaunchAgents
/com.apple.softwareupdate.plist

launchctl unload %@
RunAtLoad
```

Embedded strings



/Library/LaunchAgents

```
# FileMonitor.app/Contents/MacOS/FileMonitor -pretty
...
{
  "event" : "ES_EVENT_TYPE_NOTIFY_CREATE",
  "file" : {
    "destination" : "~/Library/LaunchAgents/
                     com.apple.softwareupdate.plist",

    "process" : {
       "pid" : 1469
       "name": softwareupdate
       "path" : "/Users/user/Desktop/softwareupdate",
    }
  }
}
```

Launch agent persistence

(~/Library/LaunchAgents/com.apple.softwareupdate.plist)

Dumping the BTM Database

…to enumerate persistently installed software

and malware!

```
# sfltool
Usage: sfltool csinfo|dumpbtm|archive|clear|resetbtm|resetlist|list|list-info [options]
```

```
# sfltool dumpbtm
…

                  UUID: EAE1D8DC-2928-47C1-BC53-6274590FE3D1
                  Name: us.zoom.ZoomDaemon
        Developer Name: Zoom Video Communications, Inc.
       Team Identifier: BJ4HAAB9B3
                  Type: legacy daemon (0x10010)
           Disposition: [enabled, allowed, visible, notified] (11)
            Identifier: us.zoom.ZoomDaemon
                   URL: file:///Library/LaunchDaemons/us.zoom.ZoomDaemon.plist
       Executable Path: /Library/PrivilegedHelperTools/us.zoom.ZoomDaemon
            Generation: 1
     Parent Identifier: Zoom Video Communications, Inc.
```

sfltool (option: dumpbtm)

The (Raw) BTM Database

…a binary plist, containing serialized objects

```
% file /var/db/com.apple.backgroundtaskmanagement/BackgroundItems-v8.btm

Apple binary property list

% plutil -p /var/db/com.apple.backgroundtaskmanagement/BackgroundItems-v8.btm
{
  "$archiver" => "NSKeyedArchiver"
  "$objects" => [
    0 => "$null"
    1 => {
      "$class" => <CFKeyedArchiverUID 0x600003e57640 [0x1e69087f8]>{value = 142}
      "itemsByUserIdentifier" => <CFKeyedArchiverUID 0x600003e57660 [0x1e69087f8]>{value = 2}
      "mdmPaloadsByIdentifier" => <CFKeyedArchiverUID 0x600003e57680 [0x1e69087f8]>{value = 140}
      "userSettingsByUserIdentifier" => <CFKeyedArchiverUID 0x600003e576a0 [0x1e69087f8]>{value = 134}
   ...
```

BTM database (contains NSKeyedArchiver)

Serialization, saves objects to a persistent archive (e.g. BTM database)

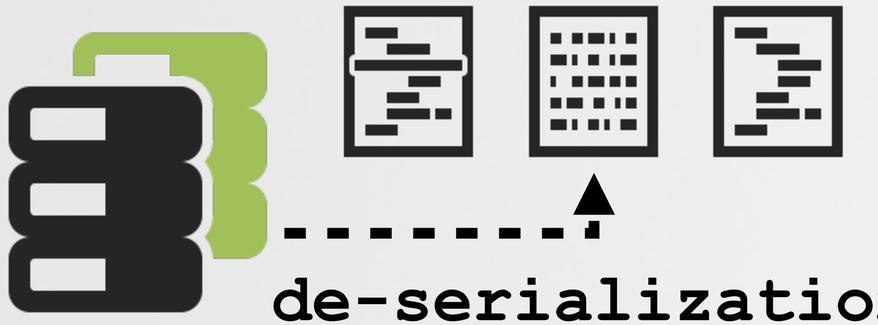
sfltool's Item De-serialization

…implemented in an "initWithCoder:" method

```
01  /* @class ItemRecord */
02  -(void *)initWithCoder:(NSCoder *)decoder {
03
04      rax = objc_opt_class(@class(NSUUID));
05      rax = [r14 decodeObjectOfClass:rax forKey:@"uuid"];
06
07      rax = objc_opt_class(@class(NSString));
08      rax = [r14 decodeObjectOfClass:rax forKey:@"executablePath"];
09
10      rax = objc_opt_class(@class(NSString));
11      rax = [r14 decodeObjectOfClass:rax forKey:@"teamIdentifier"];
12
13      …
```

method automatically invoked to de-serialize stored objects

BTM Daemon
[ItemRecord initWithCoder];



de-serialization

ItemRecord

Tag Scope

Idx	Name
173	-[ItemRecord uuid]
174	-[ItemRecord setUuid:]
175	-[ItemRecord identifier]
176	-[ItemRecord setIdentifier:]
177	-[ItemRecord disposition]
178	-[ItemRecord setDisposition:]
179	-[ItemRecord generation]
180	-[ItemRecord setGeneration:]
181	-[ItemRecord url]
182	-[ItemRecord setUrl:]
183	-[ItemRecord modificationDate]
184	-[ItemRecord setModificationDate:]
185	-[ItemRecord executablePath]
186	-[ItemRecord setExecutablePath:]
187	-[ItemRecord executableModificationDate]
188	-[ItemRecord setExecutableModificationDate:]
189	-[ItemRecord teamIdentifier]
190	-[ItemRecord setTeamIdentifier:]

'ItemRecord' properties

Parsing the BTM Database

…(re)implemented in our own code

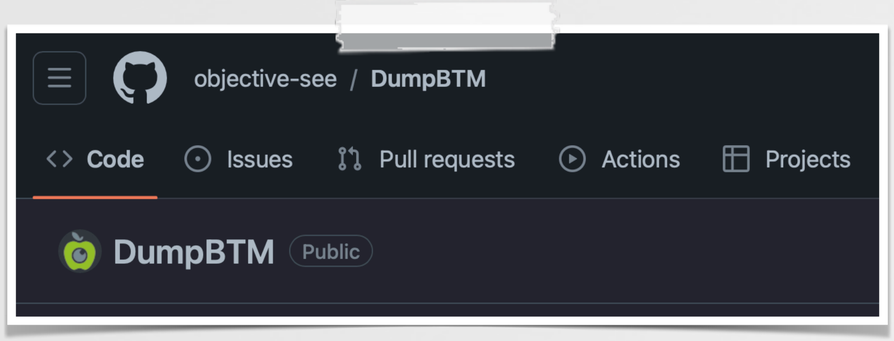
```
01  //load
02  // path set to BTM database
03  NSData* data = [NSData dataWithContentsOfURL:path options:0 error:NULL];
04
05  //init keyed unarchiver with database data
06  NSKeyedUnarchiver* keyedUnarchiver = [[NSKeyedUnarchiver alloc] initForReadingFromData:data error:NULL];
07
08  //de-serialize
09  // will trigger call to "initWithCoder:" and de-serialize all objects
10  Storage* storage = [keyedUnarchiver decodeObjectOfClass:[Storage class] forKey:@"store"];
11
12  //print out all items
13  for(NSString* key in storage.items) {
14
15      NSArray* items = storage.items[key];
16
17      for(ItemRecord* item in items){
18          printf(" #%d\n", ++itemNumber);
19          printf(" %s\n", [[item dumpVerboseDescription] UTF8String]);
20
21      }
22  }
```

my "Dump BTM" library

Parsing the BTM Database

…(re)implemented in our own code

```
01  #include "dumpBTM.h"
02
03  int main(int argc, const char * argv[]) {
04
05      //set if you want
06      // a custom btm file
07      NSURL* btmPath = nil;
08
09      //dump stdout
10      // (aka sfltool dumpbtm)
11      dumpBTM(btmPath);
12
13      //or parse into a dictionary...
14      NSDictionary* contents = parseBTM(path);
15      ...
```



github.com/objective-see/DumpBTM

"DUMP BTM" OUTPUT

...and no root access required

```
% ./dumpBTM
Opened /private/var/db/com.apple.backgroundtaskmanagement/BackgroundItems-v8.btm

=========================
 Records for UID 0 : FFFFEEEE-DDDD-CCCC-BBBB-AAAA00000000
=========================

...
 UUID:                EAE1D8DC-2928-47C1-BC53-6274590FE3D1
  Name:                us.zoom.ZoomDaemon
  Developer Name:      Zoom Video Communications, Inc.
  Team Identifier:     BJ4HAAB9B3
  Type:                legacy daemon  (0x10010)
  Disposition:         [enabled allowed visible notified] (11)
  Indentifier:         us.zoom.ZoomDaemon
  URL:                 file:///Library/LaunchDaemons/us.zoom.ZoomDaemon.plist
  Executable Path:     /Library/PrivilegedHelperTools/us.zoom.ZoomDaemon
  Generation:          2
  Assoc. Bundle IDs:   [us.zoom.xos]
  Parent Identifier:   Zoom Video Communications, Inc.
```

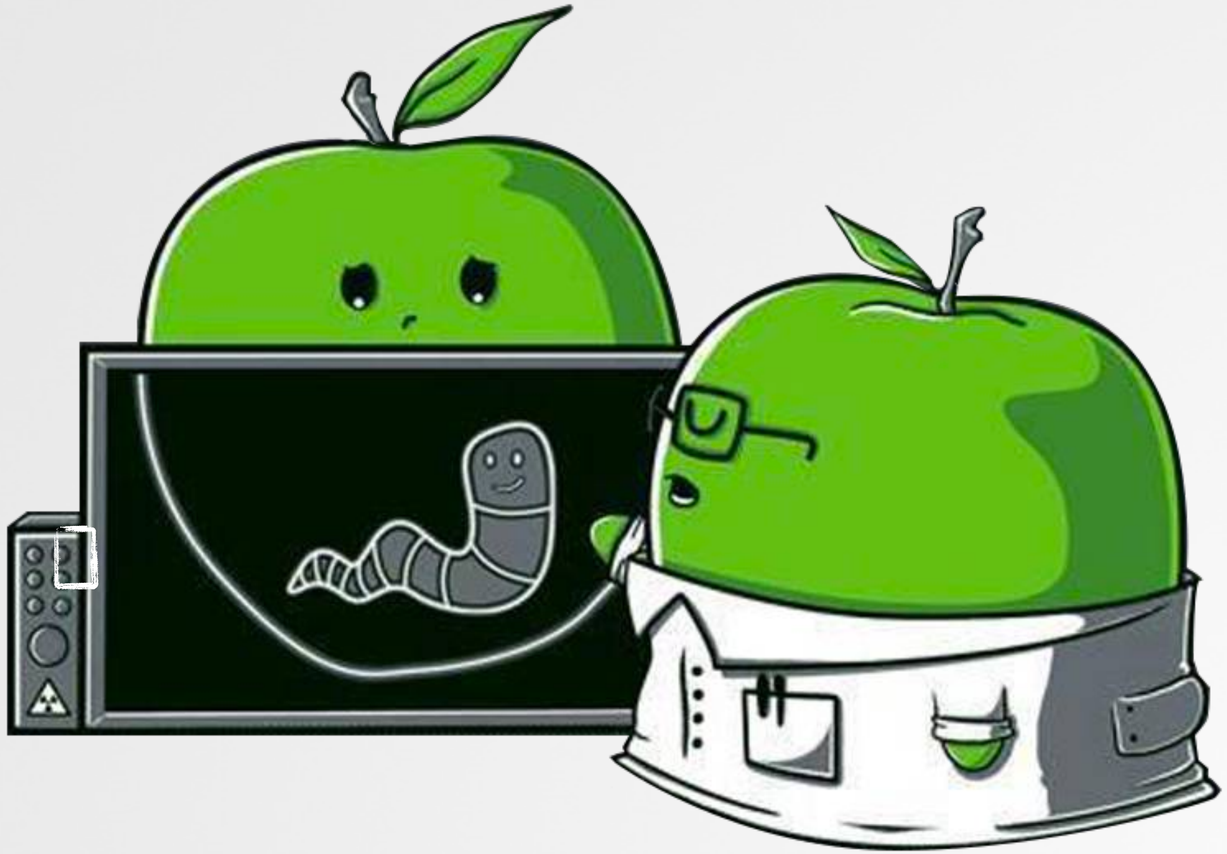
"Dump BTM" output

KnockKnock v2.5

...now with BTM item enumeration

Tools (Part II)

(proactively) detecting persistent malware via BTM



BTM Daemon's References to "Endpoint Security"

```
% strings - backgroundtaskmanagementd | grep ES

submitItemAddToES: No URL provided, cannot submit event to EndpointSecurity
submitItemAddToES: Invalid BTMItemType, cannot submit event to EndpointSecurity
```

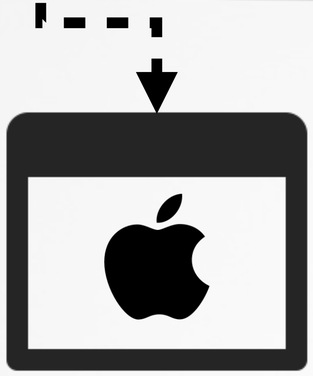
References to 0x100007b84

Search

Address	Value	
0x10000383b (-[SFLClientProcess addURL:managed:reply:] + 0x677)	call	submitItemAddToES
0x1000038ac (-[SFLClientProcess addURL:managed:reply:] + 0x6e8)	call	submitItemAddToES
0x100005053 (-[BTMService registerLaunchItemWithAuditToken:parentURL:type:relativeUR...	call	submitItemAddToES
0x100005084 (-[BTMService registerLaunchItemWithAuditToken:parentURL:type:relativeUR...	call	submitItemAddToES
0x10000519c (-[BTMService registerLaunchItemWithAuditToken:parentURL:type:relativeUR...	call	submitItemAddToES
0x1000051cc (-[BTMService registerLaunchItemWithAuditToken:parentURL:type:relativeUR...	call	submitItemAdd
0x1000065dd (-[BTMService addItem:uid:reply:] + 0x1ef)	call	submitItemAdd
0x100006611 (-[BTMService addItem:uid:reply:] + 0x223)	call	submitItemAdd

Cancel

```
01  submitItemAddToES:
02   ...
03
04   bl          ess_notify_btm_launch_item_add
```



"ess_notify_btm_launch_item_add" Google

Your search - **"ess_notify_btm_launch_item_add"** - did not match any documents.

very undocumented :/

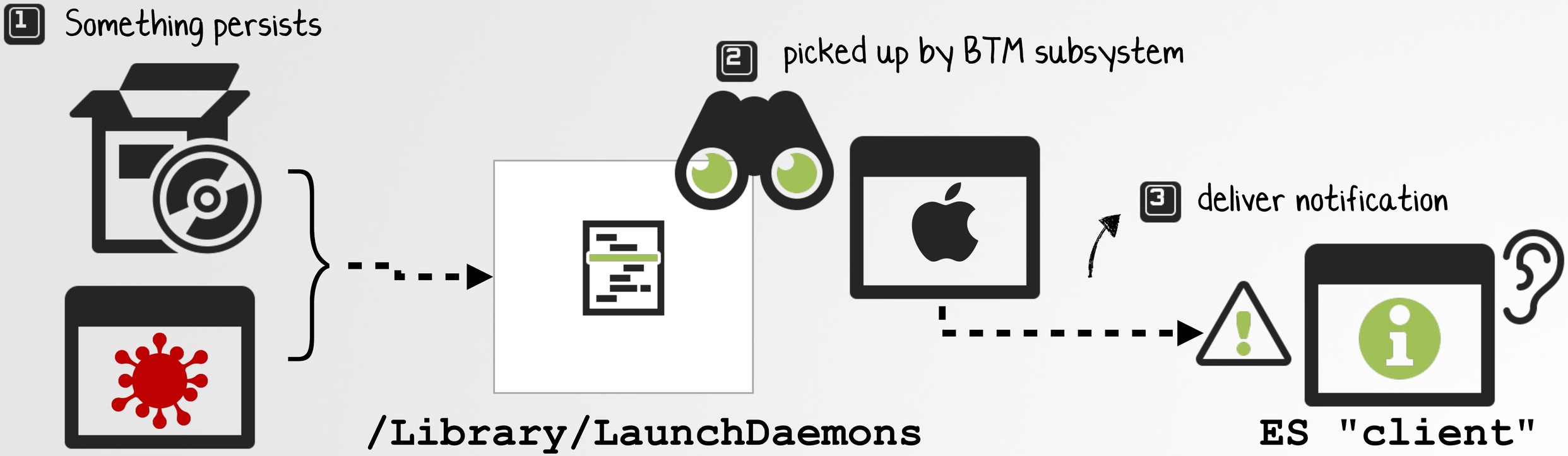
implemented in:

libEndpointSecuritySystem.dylib

Endpoint Security

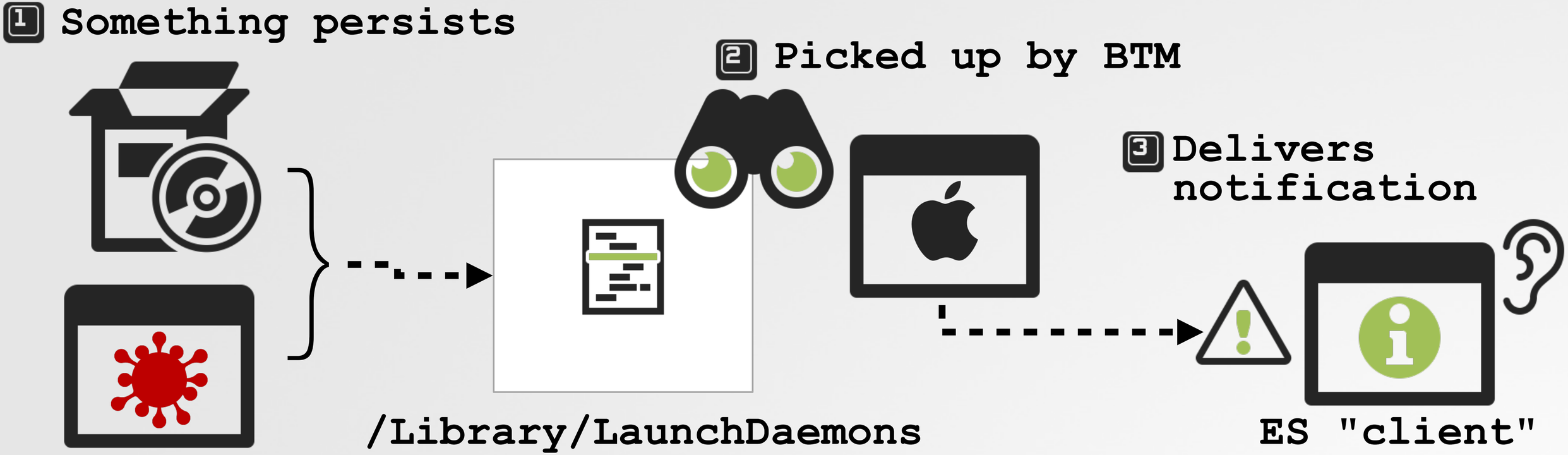
ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD event

```
/* The valid event types recognized by EndpointSecurity */
typedef enum {
...

// The following events are available beginning in macOS 13.0
...
    ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD,
    ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_REMOVE
```



Endpoint Security

ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD event

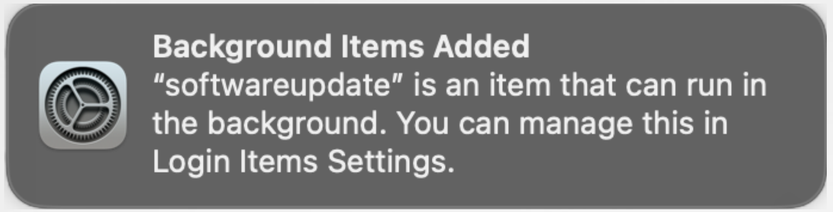



```
01  /* The valid event types recognized by EndpointSecurity */
02  typedef enum {
03  ...
04
05  // The following events are available beginning in macOS 13.0
06  ...
07      ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD,
08      ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_REMOVE
```

HOORAY, PERSISTENT MALWARE DETECTION!

...via ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD

system alert

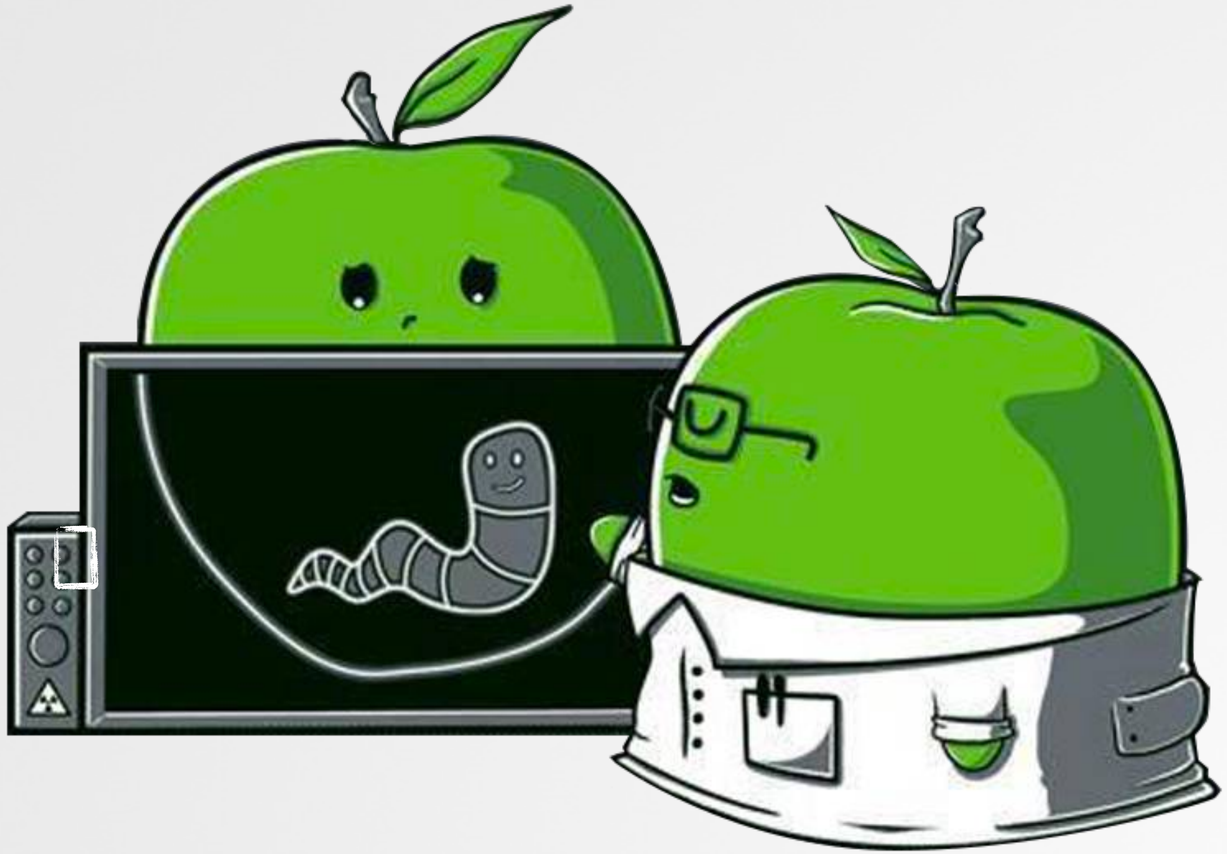



```
# ./btm_monitor
New Event: 'ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD'

Type: ES_BTM_ITEM_TYPE_AGENT

Instigator: /usr/libexec/smd

Agent exe: /Users/User/.local/softwareupdate
Agent url: /Users/User/Library/LaunchAgents/com.apple.softwareupdate.plist
```

(our) persistence detection
(OSX.DazzleSpy)

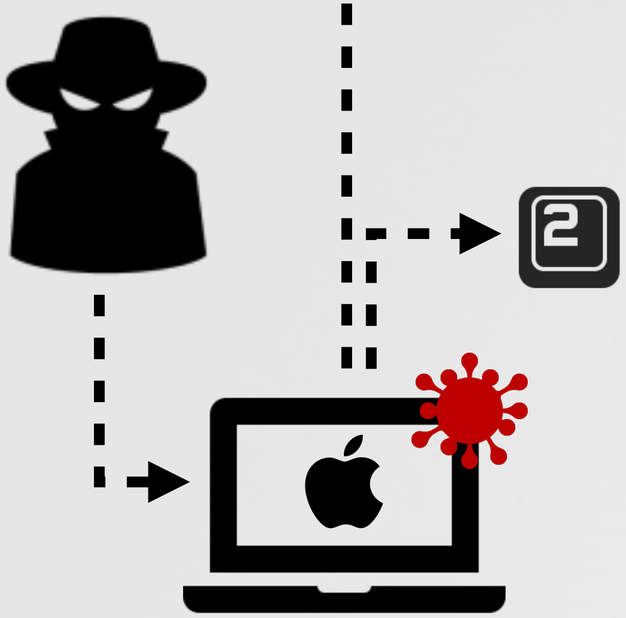
Breaking

...avoiding BTM alerts & ES messages



Hackers Hate Alerts (& Messages)

...that can give away their attack



1

Background Items Added
"softwareupdate" is an item that can run in the background. You can manage this in Login Items Settings.

2

```
# ./btm_monitor
New Event: 'ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD'

Type: ES_BTM_ITEM_TYPE_AGENT
Agent exe: /Users/User/.local/softwareupdate
Agent url: /Users/User/Library/LaunchAgents/com.apple.softwareupdate.plist
```

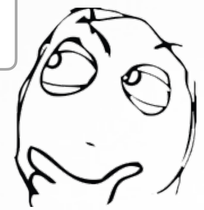
OMG, i've been hacked!!

Can we (as hackers) prevent:

A: The system alert?

B: The Endpoint Security event?

...want to persist silently!

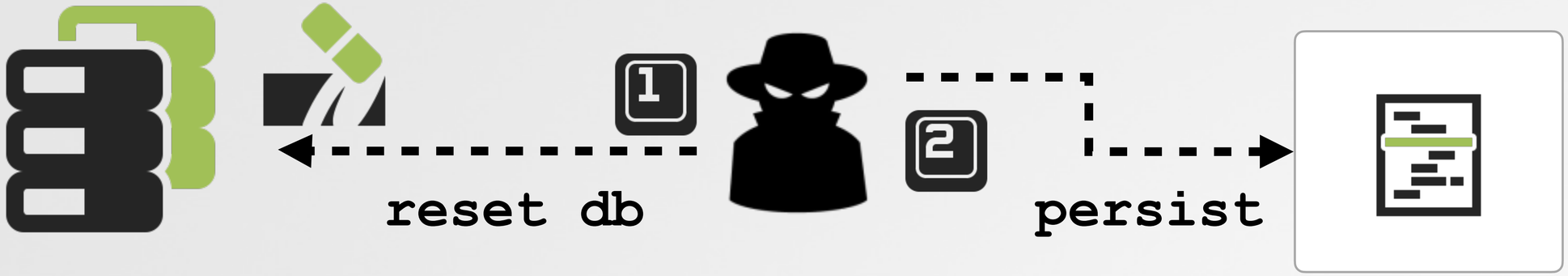


Bypass System Alert (Method 0x1)

reset the BTM database, via sfltool

note: requires root privileges

```
# sfltool resetbtm
sfltool[1261:13567] Database reset.
```

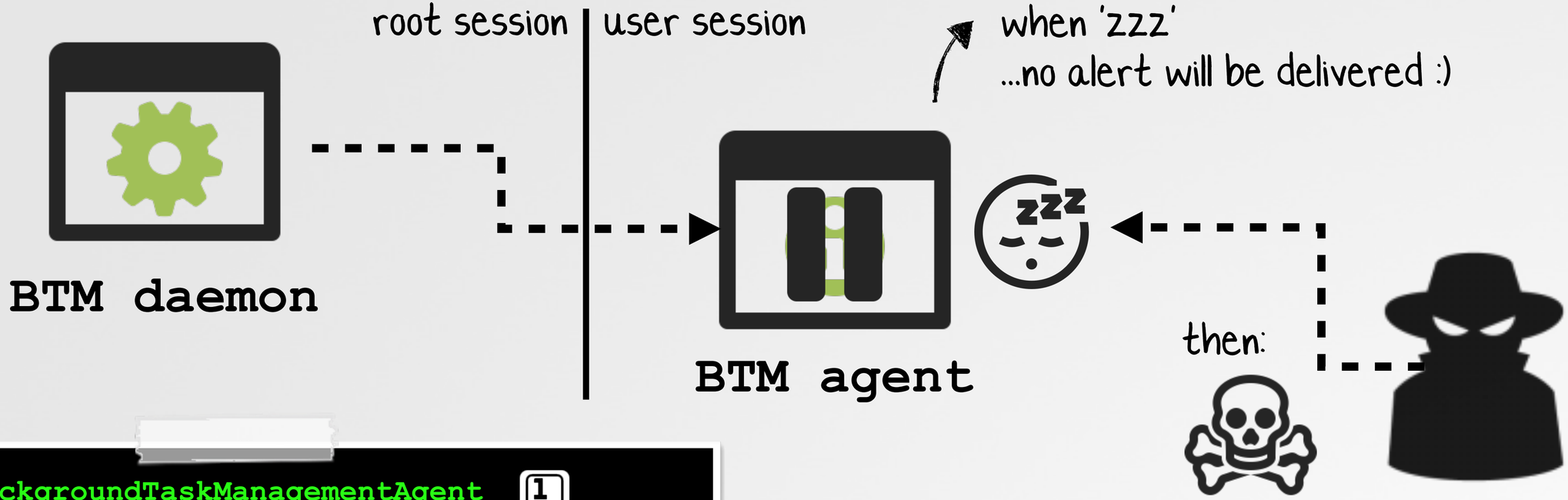



```
% log stream --debug --info --predicate "process = 'backgroundtaskmanagementd'"
backgroundtaskmanagementd: registerLaunchItem: result=no error, new item
                           disposition=[enabled, allowed, visible, not notified],
                           identifier=com.apple.softwareupdate,
                           url=file:///Users/user/Library/LaunchAgents/com.apple.softwareupdate.plist

backgroundtaskmanagementd: should post advisory=false for uid=501,
                           id=6ED3BEBC-8D60-45ED-8BCC-E0163A8AA806, item=softwareupdate
```

"should post advisory=false"

Bypass System Alert (Method 0x2)

pause, then kill the BTM agent



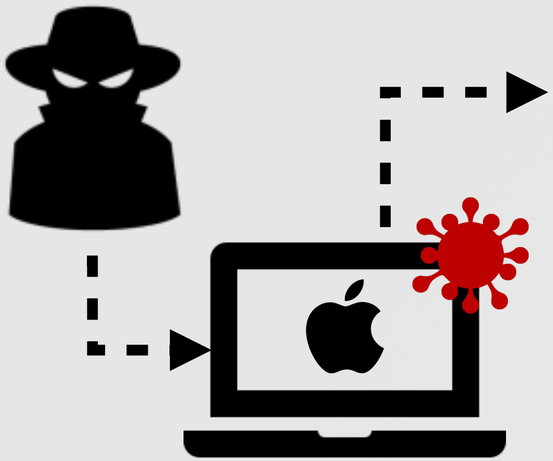

```
% pgrep BackgroundTaskManagementAgent   1
88262

% kill -SIGSTOP 88262   2

% ps -o state 88262
T

% kill -SIGKILL 88262   3
```

'T' means, stopped

1. Get pid of BTM agent
2. Send it a 'STOP' signal
3. Once persisted, kill it (to prevent alert re-delivery)

BYPASS ES EVENTS

prevent ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD



```
# ./btm_monitor
New Event: 'ES_EVENT_TYPE_NOTIFY_BTM_LAUNCH_ITEM_ADD'

item exe: /Users/User/.local/softwareupdate
item url: /Users/User/Library/LaunchAgents/com.apple.softwareupdate.plist
```

want to prevent this !

BTM daemon

```
01  def submitItemAddToES()
02    ...
03    ess_notify_btm_launch_item_add();
```

found in libEndpointSecuritySystem

```
01  def ess_notify_btm_launch_item_add()
02    ...
03    downcallNotifyOnly();
```

```
01  def downcallNotifyOnly()
02    ...
03    if(!mac_syscall("EndpointSecurity", ...)
04       os_log_impl(..., "Failed to submit event using ES syscall %d: %d %s", &var_50, 0x18);
```

mac_syscall: to the kernel

(Responsible?) Process Lookup

...via proc_find & proc_getexecutablevnode

EndpointSecurity.kext

```
% lldb
(lldb) gdb-remote 8084
...
(lldb) bt
EndpointSecurity`pdbReadAuditToken()
...
EndpointSecurity`EndpointSecurityEventManager::sendNotifyOnly()
EndpointSecurity`EndpointSecurityEventManager::sendBtmLaunchItemAdd()
EndpointSecurity`EndpointSecurityEventManager::es_policy_syscall()
kernel`hndl_unix_scall64
```

call stack

```
01  def pdbReadAuditToken()
02    ...
03    PackedDataBufferReader::read
04    ...
05    AutoReleasingProc::createFromPidVer();
06    ...
07    set_common_proc_info();
08
09    //error? no ES message delivered!
```

```
01  def AutoReleasingProc::createFromPidVer();
02    ...
03    proc* process = proc_find(<pid>);
```

```
(lldb) p *(proc*)$rdi
(proc) $1 = {
  p_pid = 866
  p_name = {'OverSight'}
```

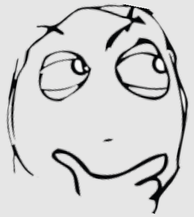
reported event (in user-mode)

```
# eslogger btm_launch_item_add
"event": {
      "btm_launch_item_add": {
       "pid": 866
       "path": "/Applications/OverSight.app
                    /Contents/MacOS/OverSight"
        ...
```

```
01  def set_common_proc_info();
02    ...
03    vnode = proc_getexecutablevnode(process);
04    if(!vnode)
05       //set error to 0x3 (ESRCH / "No Such Process")
```

(Responsible?) Process Lookup

...via proc_find & proc_getexecutablevnode



What happens if the responsible process ...has (quickly) exited already?

```
01  def AutoReleasingProc::createFromPidVer();
02    ...
03    proc* process = proc_find(NULL)
```

proc_find(NULL/0)
...will return the kernel task

```
(lldb) p *(proc*)$rdi
(proc) $1 = {
  p_pid = 0
  p_name = {'kernel_task'}
```

the kernel task: no executable vnode!

```
01  def set_common_proc_info();
02    ...
03    vnode = proc_getexecutablevnode(process)
04    if(!vnode)
05      //return ERROR 0x3: ('ESRCH' / "No Such Process")
06      rax = 0x3;
```

Error
...and no ES message delivered ! 🫣

```
% log stream ...

Error
backgroundtaskmanagementd: (libEndpointSecuritySystem.dylib)
Failed to submit event using ES syscall 20: 3 No such process
```

How to Trigger?

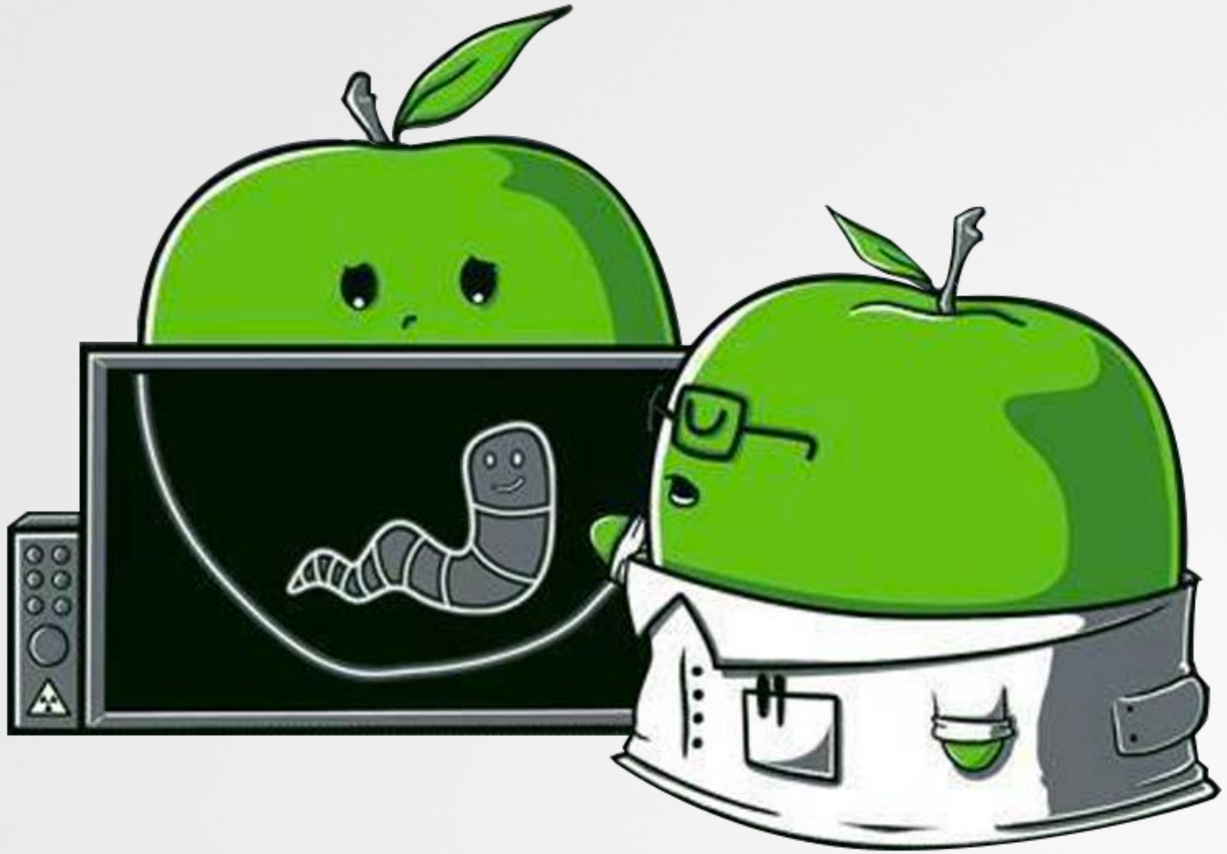
...as easy as installing BlockBlock 😅

```
01  #install logic
02  if [ "${1}" == "-install" ]; then
03   ...
04
05   cp "com.objective-see.blockblock.plist" /Library/LaunchDaemons/
06
07   echo "launch daemon installed"
```

BlockBlock Installer

Conclusions

...& take aways



Takeaways