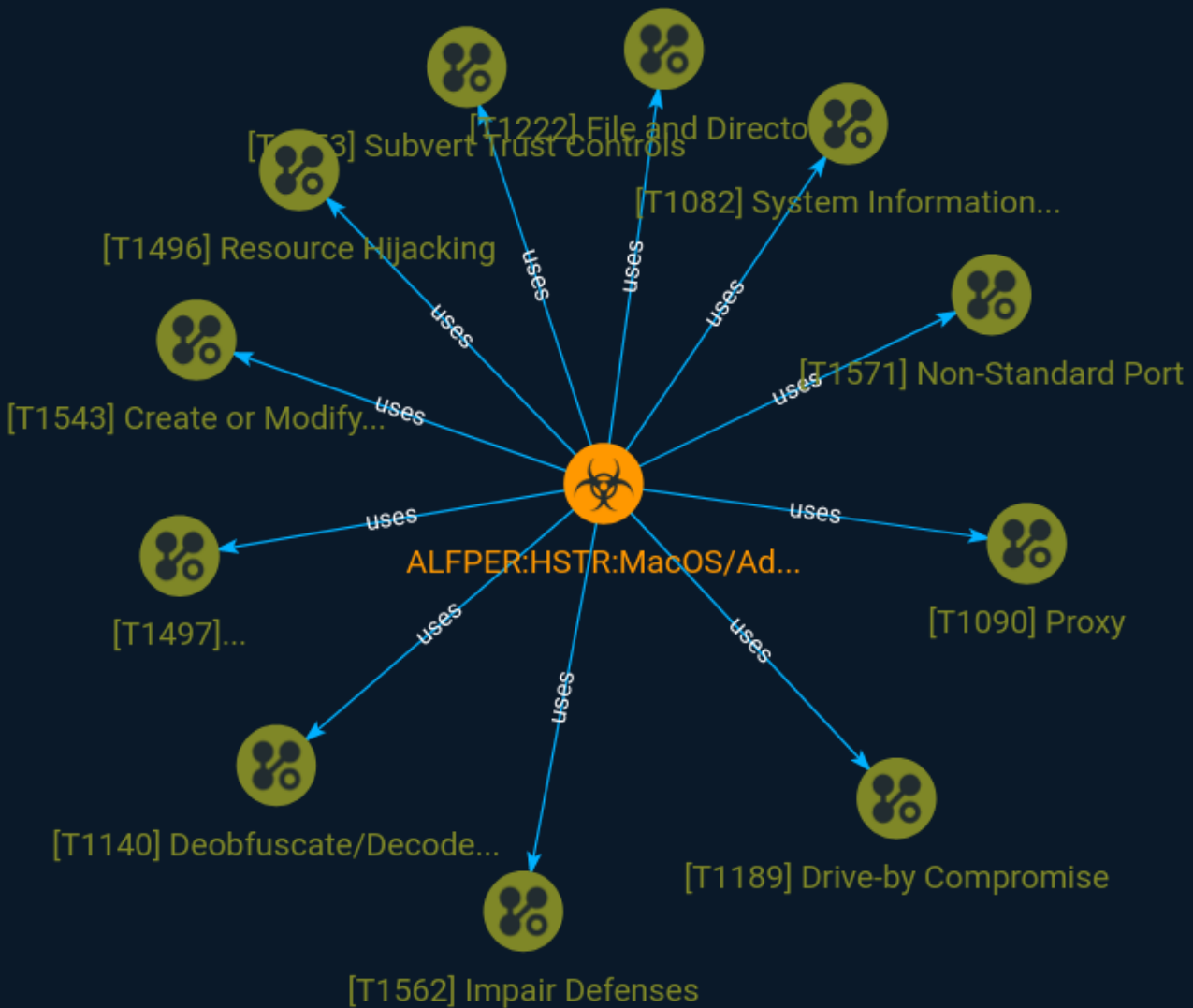




NETMANAGEIT

# Intelligence Report

## Mac systems turned into Proxy exit nodes by AdLoad



# Table of contents

---

## Overview

---

● Description	3
● Confidence	3

---

---

## Entities

---

● Malware	4
● Attack-Pattern	5

---

---

## External References

---

● External References	13
-----------------------	----

---

# Overview

## Description

AdLoad malware is still infecting Mac systems years after its first appearance. After achieving execution through Drive-by Compromise, the infected system starts operating as a proxy exit point, forwarding any HTTP(s) request it receives and routing the responses.

## Confidence

*This value represents the confidence in the correctness of the data contained within this report.*

15 / 100

# Malware

## Name

ALFPER:HSTR:MacOS/Adload

# Attack-Pattern

## Name

File and Directory Permissions Modification

## ID

T1222

## Description

Adversaries may modify file or directory permissions/attributes to evade access control lists (ACLs) and access protected files.(Citation: Hybrid Analysis Icacls1 June 2018)(Citation: Hybrid Analysis Icacls2 May 2018) File and directory permissions are commonly managed by ACLs configured by the file or directory owner, or users with the appropriate permissions. File and directory ACL implementations vary by platform, but generally explicitly designate which users or groups can perform which actions (read, write, execute, etc.). Modifications may include changing specific access rights, which may require taking ownership of a file or directory and/or elevated permissions depending on the file or directory's existing permissions. This may enable malicious activity such as modifying, replacing, or deleting specific files or directories. Specific file and directory modifications may be a required step for many techniques, such as establishing Persistence via [Accessibility Features](<https://attack.mitre.org/techniques/T1546/008>), [Boot or Logon Initialization Scripts](<https://attack.mitre.org/techniques/T1037>), [Unix Shell Configuration Modification](<https://attack.mitre.org/techniques/T1546/004>), or tainting/hijacking other instrumental binary/configuration files via [Hijack Execution Flow](<https://attack.mitre.org/techniques/T1574>). Adversaries may also change permissions of symbolic links. For example, malware (particularly ransomware) may modify symbolic links and associated settings to enable access to files from local shortcuts with remote paths.(Citation: new\_rust\_based\_ransomware)(Citation: bad\_luck\_blackcat)(Citation: falconoverwatch\_blackcat\_attack)(Citation: blackmatter\_blackcat)(Citation: fsutil\_behavior)

**Name**

Create or Modify System Process

**ID**

T1543

**Description**

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services.(Citation: TechNet Services) On macOS, launchd processes known as [Launch Daemon](<https://attack.mitre.org/techniques/T1543/004>) and [Launch Agent](<https://attack.mitre.org/techniques/T1543/001>) are run to finish system initialization and load user specific parameters.(Citation: AppleDocs Launch Agent Daemons) Adversaries may install new services, daemons, or agents that can be configured to execute at startup or a repeatable interval in order to establish persistence. Similarly, adversaries may modify existing services, daemons, or agents to achieve the same effect. Services, daemons, or agents may be created with administrator privileges but executed under root/SYSTEM privileges. Adversaries may leverage this functionality to create or modify system processes in order to escalate privileges.(Citation: OSX Malware Detection)

**Name**

Subvert Trust Controls

**ID**

T1553

**Description**

Adversaries may undermine security controls that will either warn users of untrusted activity or prevent execution of untrusted programs. Operating systems and security products may contain mechanisms to identify programs or websites as possessing some

level of trust. Examples of such features would include a program being allowed to run because it is signed by a valid code signing certificate, a program prompting the user with a warning because it has an attribute set from being downloaded from the Internet, or getting an indication that you are about to connect to an untrusted site. Adversaries may attempt to subvert these trust mechanisms. The method adversaries use will depend on the specific mechanism they seek to subvert. Adversaries may conduct [File and Directory Permissions Modification](<https://attack.mitre.org/techniques/T1222>) or [Modify Registry](<https://attack.mitre.org/techniques/T1112>) in support of subverting these controls. (Citation: SpectorOps Subverting Trust Sept 2017) Adversaries may also create or steal code signing certificates to acquire trust on target systems.(Citation: Securelist Digital Certificates)(Citation: Symantec Digital Certificates)

**Name**

Non-Standard Port

**ID**

T1571

**Description**

Adversaries may communicate using a protocol and port pairing that are typically not associated. For example, HTTPS over port 8088(Citation: Symantec Elfin Mar 2019) or port 587(Citation: Fortinet Agent Tesla April 2018) as opposed to the traditional port 443. Adversaries may make changes to the standard port used by a protocol to bypass filtering or muddle analysis/parsing of network data. Adversaries may also make changes to victim systems to abuse non-standard ports. For example, Registry keys and other configuration settings can be used to modify protocol and port pairings.(Citation: change\_rdp\_port\_conti)

**Name**

Virtualization/Sandbox Evasion

**ID**

T1497

**Description**

Adversaries may employ various means to detect and avoid virtualization and analysis environments. This may include changing behaviors based on the results of checks for the presence of artifacts indicative of a virtual machine environment (VME) or sandbox. If the adversary detects a VME, they may alter their malware to disengage from the victim or conceal the core functions of the implant. They may also search for VME artifacts before dropping secondary or additional payloads. Adversaries may use the information learned from [Virtualization/Sandbox Evasion](<https://attack.mitre.org/techniques/T1497>) during automated discovery to shape follow-on behaviors.(Citation: Deloitte Environment Awareness) Adversaries may use several methods to accomplish [Virtualization/Sandbox Evasion](<https://attack.mitre.org/techniques/T1497>) such as checking for security monitoring tools (e.g., Sysinternals, Wireshark, etc.) or other system artifacts associated with analysis or virtualization. Adversaries may also check for legitimate user activity to help determine if it is in an analysis environment. Additional methods include use of sleep timers or loops within malware code to avoid operating within a temporary sandbox. (Citation: Unit 42 Pirpi July 2015)

**Name**

Proxy

**ID**

T1090

**Description**

Adversaries may use a connection proxy to direct network traffic between systems or act as an intermediary for network communications to a command and control server to avoid direct connections to their infrastructure. Many tools exist that enable traffic redirection through proxies or port redirection, including [HTRAN](<https://attack.mitre.org/software/S0040>), ZXProxy, and ZXPortMap. (Citation: Trend Micro APT Attack Tools) Adversaries use these types of proxies to manage command and control communications, reduce the number of simultaneous outbound network connections, provide resiliency in the face of connection loss, or to ride over existing trusted communications paths between victims to avoid suspicion. Adversaries may chain together multiple proxies to further disguise the source of malicious traffic. Adversaries can also take advantage of routing schemes in Content Delivery Networks (CDNs) to proxy command and control traffic.



**Name**

Impair Defenses

**ID**

T1562

**Description**

Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators. Adversaries may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out of a computer or stopping it from being shut down. These restrictions can further enable malicious operations as well as the continued propagation of incidents.(Citation: Emotet shutdown) Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

**Name**

Resource Hijacking

**ID**

T1496

**Description**

Adversaries may leverage the resources of co-opted systems in order to solve resource intensive problems, which may impact system and/or hosted service availability. One common purpose for Resource Hijacking is to validate transactions of cryptocurrency networks and earn virtual currency. Adversaries may consume enough system resources to negatively impact and/or cause affected machines to become unresponsive.(Citation: Kaspersky Lazarus Under The Hood Blog 2017) Servers and cloud-based systems are

common targets because of the high potential for available resources, but user endpoint systems may also be compromised and used for Resource Hijacking and cryptocurrency mining.(Citation: CloudSploit - Unused AWS Regions) Containerized environments may also be targeted due to the ease of deployment via exposed APIs and the potential for scaling mining activities by deploying or compromising multiple containers within an environment or cluster.(Citation: Unit 42 Hildegard Malware)(Citation: Trend Micro Exposed Docker APIs) Additionally, some cryptocurrency mining malware identify then kill off processes for competing malware to ensure it's not competing for resources.(Citation: Trend Micro War of Crypto Miners) Adversaries may also use malware that leverages a system's network bandwidth as part of a botnet in order to facilitate [Network Denial of Service](<https://attack.mitre.org/techniques/T1498>) campaigns and/or to seed malicious torrents.(Citation: GoBotKR)

**Name**

Drive-by Compromise

**ID**

T1189

**Description**

Adversaries may gain access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is typically targeted for exploitation, but adversaries may also use compromised websites for non-exploitation behavior such as acquiring [Application Access Token](<https://attack.mitre.org/techniques/T1550/001>). Multiple ways of delivering exploit code to a browser exist (i.e., [Drive-by Target](<https://attack.mitre.org/techniques/T1608/004>)), including: \* A legitimate website is compromised where adversaries have injected some form of malicious code such as JavaScript, iFrames, and cross-site scripting \* Script files served to a legitimate website from a publicly writeable cloud storage bucket are modified by an adversary \* Malicious ads are paid for and served through legitimate ad providers (i.e., [Malvertising](<https://attack.mitre.org/techniques/T1583/008>)) \* Built-in web application interfaces are leveraged for the insertion of any other kind of object that can be used to display web content or contain a script that executes on the visiting client (e.g. forum posts, comments, and other user controllable web content). Often the website used by an adversary is one visited by a specific community, such as government, a particular industry, or region, where the goal is to compromise a specific user or set of users based on a shared interest. This kind of targeted campaign is often referred to a strategic web compromise or watering hole attack. There are several known examples of this occurring.(Citation: Shadowserver

Strategic Web Compromise) Typical drive-by compromise process: 1. A user visits a website that is used to host the adversary controlled content. 2. Scripts automatically execute, typically searching versions of the browser and plugins for a potentially vulnerable version. \* The user may be required to assist in this process by enabling scripting or active website components and ignoring warning dialog boxes. 3. Upon finding a vulnerable version, exploit code is delivered to the browser. 4. If exploitation is successful, then it will give the adversary code execution on the user's system unless other protections are in place. \* In some cases a second visit to the website after the initial scan is required before exploit code is delivered. Unlike [Exploit Public-Facing Application](<https://attack.mitre.org/techniques/T1190>), the focus of this technique is to exploit software on a client endpoint upon visiting a website. This will commonly give an adversary access to systems on the internal network instead of external systems that may be in a DMZ. Adversaries may also use compromised websites to deliver a user to a malicious application designed to [Steal Application Access Token](<https://attack.mitre.org/techniques/T1528>), like OAuth tokens, to gain access to protected applications and information. These malicious applications have been delivered through popups on legitimate websites.(Citation: Volexity OceanLotus Nov 2017)

**Name**

Deobfuscate/Decode Files or Information

**ID**

T1140

**Description**

Adversaries may use [Obfuscated Files or Information](<https://attack.mitre.org/techniques/T1027>) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is the use of [certutil](<https://attack.mitre.org/software/S0160>) to decode a remote access tool portable executable file that has been hidden inside a certificate file.(Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b`` command to reassemble binary fragments into a malicious payload.(Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [User Execution](<https://attack.mitre.org/techniques/T1204>). The user may also be required to input a password to open a password protected compressed/

encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016)

**Name**

System Information Discovery

**ID**

T1082

**Description**

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](<https://attack.mitre.org/techniques/T1082>) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Tools such as [Systeminfo](<https://attack.mitre.org/software/S0096>) can be used to gather detailed system information. If running with privileged access, a breakdown of system data can be gathered through the `systemsetup` configuration tool on macOS. As an example, adversaries with user-level access can execute the `df -aH` command to obtain currently mounted disks and associated freely available space. Adversaries may also leverage a [Network Device CLI](<https://attack.mitre.org/techniques/T1059/008>) on network devices to gather detailed system information (e.g. `show version`). (Citation: US-CERT-TA18-106A) [System Information Discovery](<https://attack.mitre.org/techniques/T1082>) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment. (Citation: OSX.FairyTale) (Citation: 20 macOS Common Tools and Techniques) Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine. (Citation: Amazon Describe Instance) (Citation: Google Instances Resource) (Citation: Microsoft Virtual Machine API)

# External References

- 
- <https://otx.alienvault.com/pulse/64a442b7bdaa1bd2ea1ebfd4>