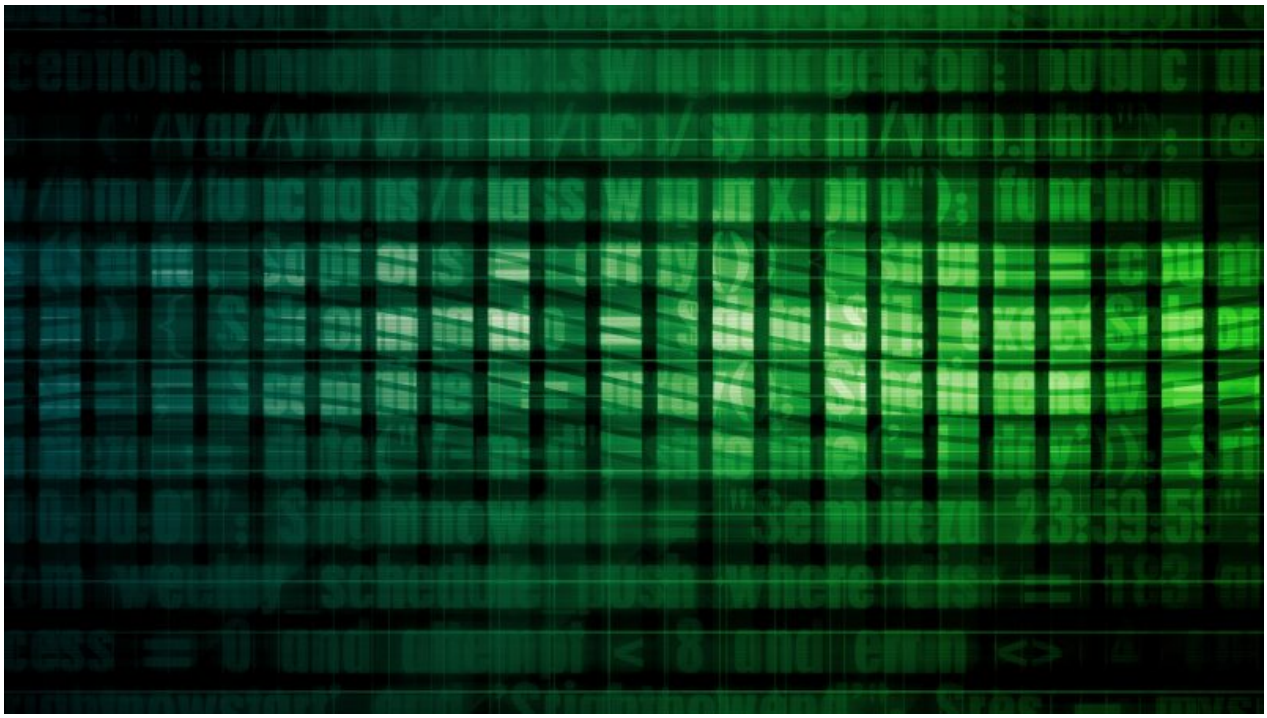
What did DeathStalker hide between two ferns?

SL securelist.com/what-did-deathstalker-hide-between-two-ferns/99616/



Authors

 Pierre Delcher

DeathStalker is a threat actor that's been active since at least 2012, and we exposed most of their past activities in a previous article, as well as during a GREAT Ideas conference in August 2020. The actor drew our attention in 2018 because of distinctive attack characteristics that didn't fit in with the usual cybercrime or state-sponsored activities, leading us to believe DeathStalker is a hack-for-hire group..

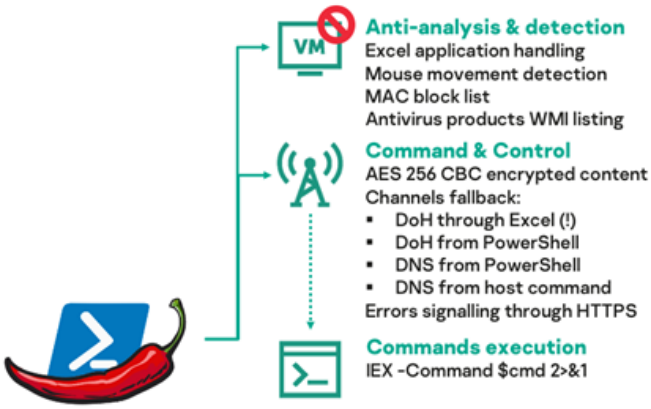
DeathStalker has leveraged several malware strains and delivery chains over the years, from the Python- and VisualBasic-based Janicab to the PowerShell-based Powersing and the JavaScript-based Evilnum. The actor consistently used what we call "dead-drop resolvers" (DDRs), which is obfuscated content hosted on major public web services like YouTube, Twitter or Reddit; once decoded by malware this content reveals a command-and-control (C2) server address. DeathStalker also consistently leveraged anti-detection and antivirus evasion techniques, as well as intricate delivery chains that drop lots of files to the target's filesystems. To kick-start an infection, DeathStalker usually relies on spear-phishing emails with attachments, or links to public file sharing services, as well as script execution based on Windows shortcuts. We have identified how DeathStalker's malware compromises in clusters or targets various types of entities in all parts of the world, with a possible focus on law and consultancy offices, as well as FINTECH companies, but without a clearly identifiable or consistent interest. The targeting does not seem to be politically or strategically defined and doesn't appear to be the usual financially motived crime. Because of this, we conclude that DeathStalker is a cyber-mercenary organization.

While tracking DeathStalker's Powersing-based activities in May 2020, we detected a previously unknown implant that leveraged DNS over HTTPS as a C2 channel, as well as parts of its delivery chain. We named this new malware PowerPepper. We first spotted a variant of PowerPepper in the wild in mid-July 2020, dropped from a Word document that had been submitted on a public multiscanner service. Since then, the PowerPepper implant and the associated delivery chain has been continuously operating and developing.

Meet PowerPepper: the spicy implant that your bland scripts setup needed

PowerPepper implant

PowerPepper is a Windows in-memory PowerShell backdoor that can execute remotely sent shell commands. In strict accordance with DeathStalker's traditions, the implant will try to evade detection or sandboxes execution with various tricks such as detecting mouse movements, filtering the client's MAC addresses, and adapting its execution flow depending on detected antivirus products.



The implant's C2 logic stands out, as it is based on communications via DNS over HTTPS (DoH), using CloudFlare responders. PowerPepper first tries to leverage Microsoft's Excel as a Web client to send DoH requests to a C2 server, but will fall back to PowerShell's standard web client, and ultimately to regular DNS communications, if messages cannot get through.

C2 communications content between the implant and servers is encrypted. We noticed that PowerPepper and the previously described Powersing use an almost identical PowerShell implementation of AES encryption, with only the AES padding mode and a function input format being changed.

PowerPepper DNS command and control

PowerPepper regularly polls a C2 server for commands to execute. In order to do so, the implant sends TXT-type DNS requests (with DoH or plain DNS requests if the former fails) to the name servers (NS) that are associated with a malicious C2 domain name. If the target which runs the implant is validated (we cover that later), the server replies with a DNS response, embedding an encrypted command. Both requests and responses contain patterns that can be easily detected with network intrusion detection systems, but the patterns have been changed across implant variants.

The command execution results are sent back to the server through a batch of variable-length A-type DNS requests, where queried hostnames contain an identifier, data length, and encrypted data.

Visual Basic

```
# Command result feedback initialization DNS request hostname:

<identifier>.be.0.0.1.0.0.0.0.<domain>

# Command result feedback data slices DNS requests hostnames:
<identifier>.ef.1.0.1.3.BDA2ADBE3C79C9EF6630.DDD4B8D4504FEC348C9C.2F53BFB60C1890585CF7.<domain>
<identifier>.ef.2.0.1.3.72DE8DDB802C4829B2DE.40CB7163E83DE0B4A002.6B6C2E555A931721A525.<domain>
<identifier>.ef.3.0.1.2.1699380DBABAB113D32B.7869501E5FEDD524304B.0.<domain>

# Command result feedback termination DNS request hostname:

<identifier>.ca.4.0.1.00.0.0.0.<domain>
```

During the course of our investigations, we noticed that the PowerPepper C2 name servers were actually open DNS resolvers that always resolved arbitrary hostnames with the same IP addresses: 128.49.4.4 (a US Navy-owned server), 91.214.6.100 and 91.214.6.101 (HSBC UK-owned servers). Using this fact and historical reverse DNS resolutions data, we have been able to preemptively identify the PowerPepper C2 domains.

PowerPepper signaling and target validation

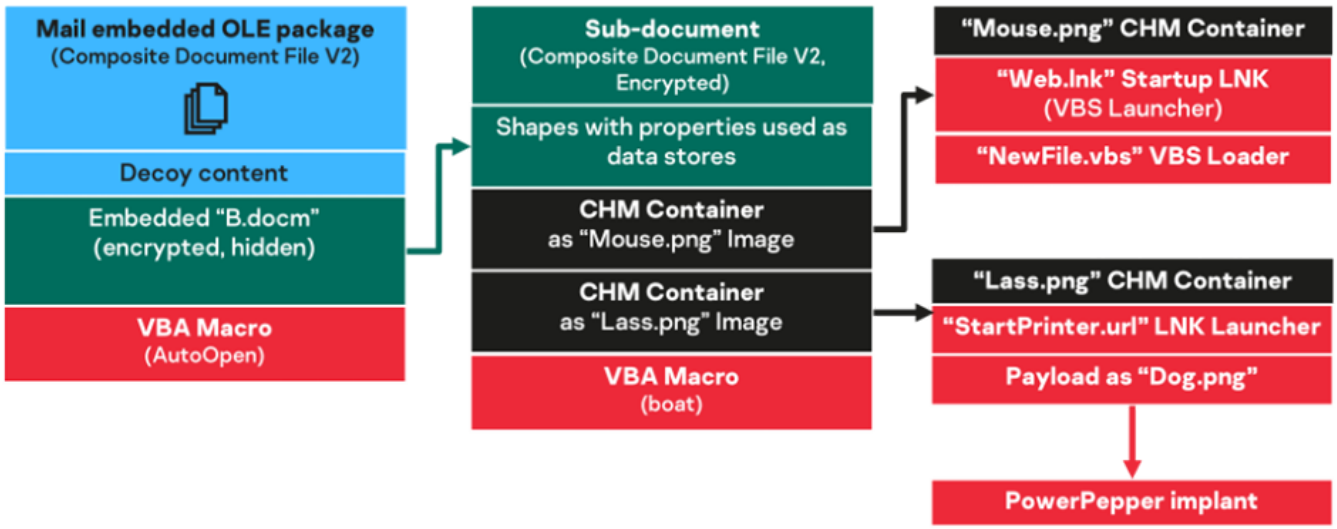
On top of the DNS C2 communication logic, PowerPepper also signals successful implant startup and execution flow errors to a Python backend, through HTTPS. Such signaling enables target validation and implant execution logging, while preventing researchers from interacting further with the PowerPepper malicious C2 name servers. It has also been used directly from some of the malicious documents that were involved in PowerPepper delivery, through the "Links to Files" feature in Office documents.

The signaling Python backends were hosted on a public and legitimate content hosting web service named PythonAnywhere that allows users to build websites. The discovered Python backend endpoints were shut down by PythonAnywhere in coordination with us. As a result, DeathStalker tried to adapt the signaling feature by removing it from most PowerPepper delivery documents (but keeping it in the implant itself), and by adding a legitimate but compromised WordPress website as a reverse-proxy between implants and backends.

PowerPepper delivery chains: a surprising journey into mercenary tricks, from Russian dolls to plant-covered steganography

The macro-based delivery chain: when you are way too much into this whole Russian dolls idea

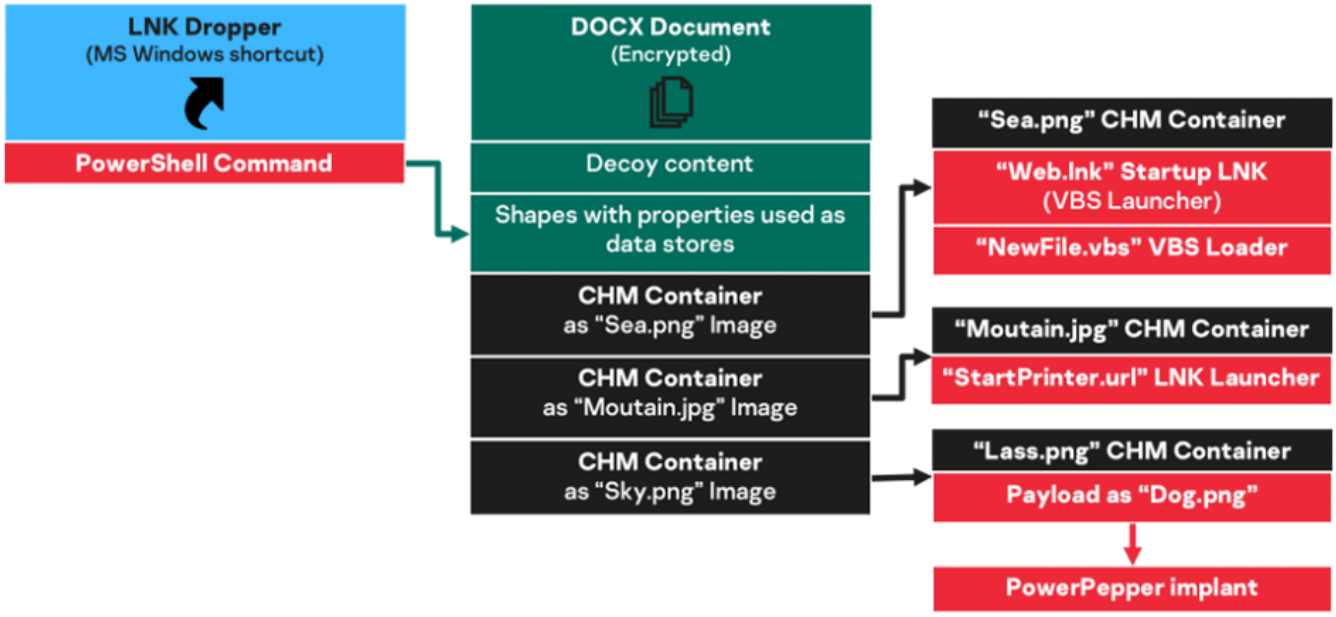
The first type of PowerPepper delivery (or infection) chain we encountered, back in July 2020, is based on a malicious Word document. Although we couldn't confirm how this document had been distributed to targets, the infection trails and documents we analyzed showed that the item is either embedded as a spear-phishing email body, or downloaded from a malicious link in a spear-phishing email. This infection chain varied slightly between July and November 2020: some dropped file names, integrated code or remote links changed, but the logic stayed the same.



We won't dive deep into the details of the delivery workflow, as the main tricks are addressed later. It should, however, be noted that the delivery chain is based on a monolithic document that embeds all required malicious items. Notably, this document contains decoy content, and the malicious logic is handled by Visual Basic for Application (VBA) macros, which ultimately run PowerPepper and set up its persistence.

The LNK-based delivery chain: your direct shortcut to spiciness

This infection chain is based on a Windows shortcut file, with a misleading .docx.lnk double extension, and constitutes a more modular approach to PowerPepper delivery.



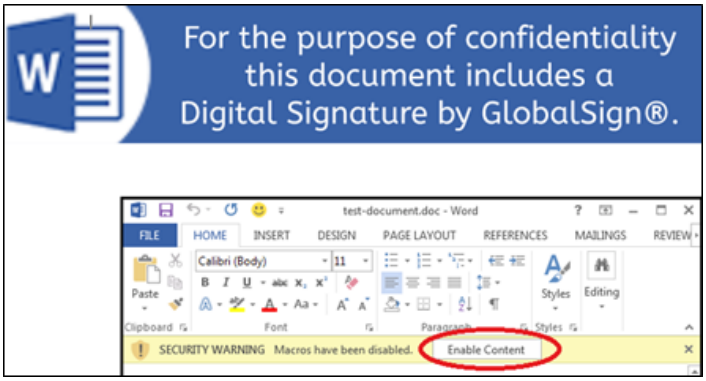
The delivery chain is very similar to the macro-based one, but implements two major changes:

- the malicious macros logic is moved to malicious PowerShell scripts, and the first one is directly embedded in the shortcut file, so there are no more VBA macros;
- the Word document from this chain is just a decoy and malicious files storage pack, and is downloaded from a remote location (a public file sharing service) instead of directly embedded somewhere.

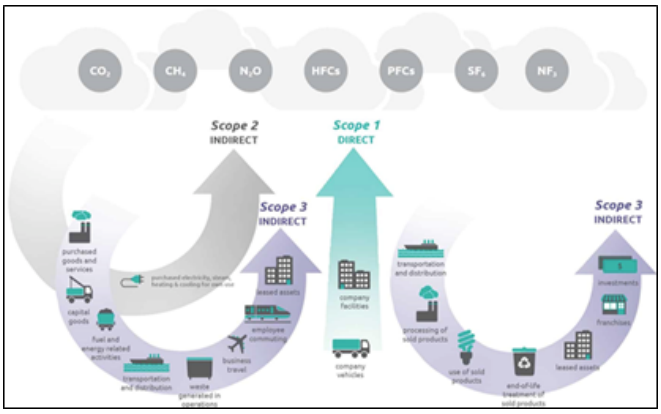
The malicious LNK files were most likely distributed as ZIP attachments within spear-phishing emails and, of course, the files dropped from this delivery chain differ across variants as well.

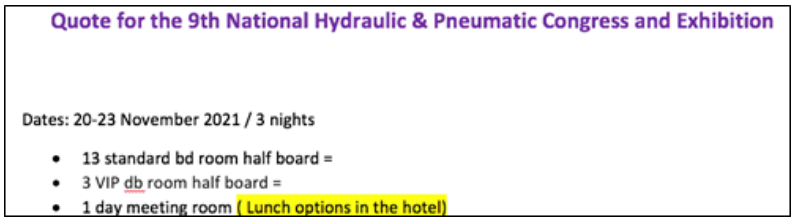
A quick look at the decoy contents

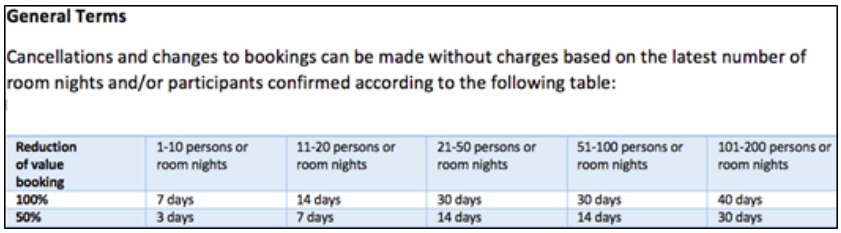
Some malicious documents that we managed to retrieve contained a social engineering banner asking users to enable macros execution. This explains how the malicious logic from the macro-based delivery chain could actually be triggered when macros are disabled by default on most modern Office settings.



The decoy contents we retrieved varied: the first we found in the wild were about carbon emissions regulations, but we also identified a fake travel booking form for a very specific event that's planned next year in Turkey, and of course some are about the coronavirus.







General Terms

Cancellations and changes to bookings can be made without charges based on the latest number of room nights and/or participants confirmed according to the following table:

Reduction of value booking	1-10 persons or room nights	11-20 persons or room nights	21-50 persons or room nights	51-100 persons or room nights	101-200 persons or room nights
100%	7 days	14 days	30 days	30 days	40 days
50%	3 days	7 days	14 days	14 days	30 days

We were able to link most of the decoy contents back to the original contents published on the internet by their initial authors, meaning DeathStalker did not craft them, but instead picked out appropriate ready-made material that was available on the internet. One of the decoy components impersonated a legitimate travel agent but included altered contact details.

A compilation of PowerPepper tricks

PowerPepper delivery chains leverage a lot of obfuscation, execution and masquerading tricks to hinder detection, or deceive targets that are curious about what is happening on their computers. So, we thought we should describe some.

Trick #1: hide things in Word embedded shape properties (and make macro comments fun again)

DeathStalker hides strings in Word embedded shape and object (OLE packages) properties, like the “hyperlink” property, to obfuscate the malicious execution workflow, as well as reconstruct and execute commands or scripts.

Visual Basic

```
1   bell = "JohnSnow123"
2   …
3   Documents.Open FileName:=best & FName, PasswordDocument:=CStr(bell), Visible:=False
4   Documents.Item(FName).Activate
5   With Application:
6           .Run "boat", belt
7   …
8   ' this function is totally legit and if you are an av you should totally let us pass
9   Function boat(both)
10  …
11  ' checks if the type is 7
12  If .Type = 7 Then
13  …
14  If .OLEFormat.Application = "Microsoft Word" And .OLEFormat.ClassType = "Package" Then
15          band = Split(.Hyperlink.Address, "ps://")
16          …
17          ball = ball & band(1)
```

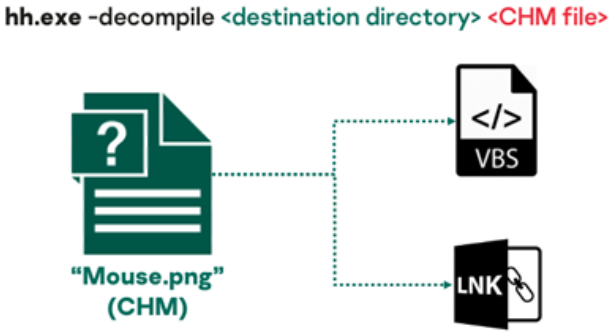
Notably, these properties are leveraged as a second stage PowerShell script in the LNK-based delivery chain: the first stage PowerShell script, which is embedded in a malicious LNK file, will parse downloaded Word document contents to extract and run a second PowerShell script. These property artifacts can also contain parts of URLs, dropped files

paths, or commands that are directly leveraged by macros in the macro-based delivery chain.

We can also see from the code extract above that DeathStalker uses macros to open another subdocument that is embedded in the first malicious document from the macro-based delivery chain. Last but not least, the comments are very helpful.

Trick #2: use Windows Compiled HTML Help (CHM) files as archives for malicious files

In the course of their PowerPepper delivery workflows, DeathStalker leverages CHM files as containers to better evade detection, and uses a Windows built-in tool called “hh” to unpack content, from VBA macros or an LNK-embedded PowerShell script.



All the files that are dropped on targeted computers from delivery chains and that are necessary to run PowerPepper are contained in these archives. The CHM files are embedded in the malicious Word (sub)document of the delivery chains.

Trick #3: masquerade and obfuscate persistent files

DeathStalker uses a Visual Basic Script (VBS) loader to start PowerPepper execution. The loader is launched immediately after delivery, and then at each computer startup, thanks to a companion launcher shortcut which is placed in a Windows startup folder.

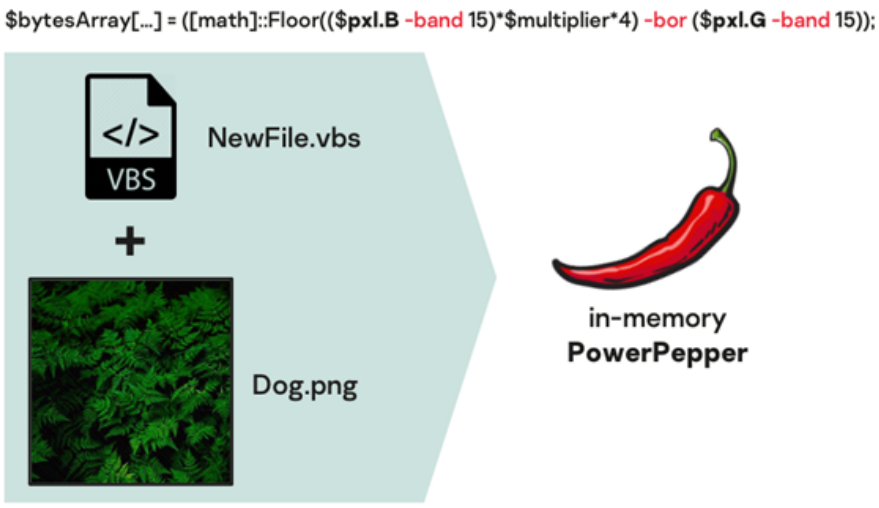
Visual Basic

```
1   ' Copyright (c) GlobalSign Corporation. All rights reserved.
2   '
3   ' Abstract:
4   ' licenseverification.vbs - Verify the GlobalSign software
5   '
6   ' Usage:
7   ' licenseverify [-software]
8   …
9   const L_Help_Help_General05_Text   = "-a     - add a port"
10  const L_Help_Help_General06_Text   = "-d     - delete the specified port"
11  …
12  const L_Help_Help_General34_Text   = "417079070765161B1C0eeeeeef610520C0F69331…
13  …
14  CreateObject(DelPort(L_Text_Msg_Port01_Text)).Run …+DelPort(L_Help_Help_General34_Text & "7260D3…
```

This VBS loader masquerades as a GlobalSign verification tool with comments as well as deceptive variables or help strings. Furthermore, the script's malicious content is obfuscated by a custom encryption function.

Trick #4: hide your implant between two ferns…

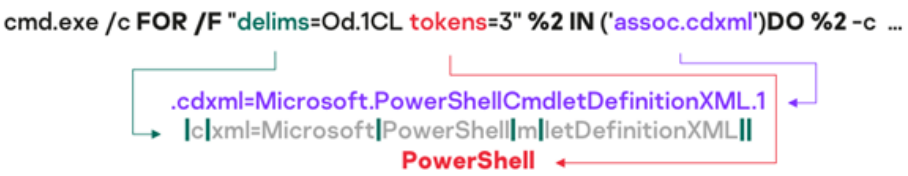
And here come our plants…. The previously described VBS loader will basically do one thing: deobfuscate and run a PowerShell script against a picture file that was dropped earlier from the delivery chain.



But the picture is actually a steganography image (of ferns…) that will be decoded by the VBS loader-embedded script, and contains the PowerPepper implant. In the first delivery chains that were discovered, the steganography image actually displayed peppers, hence the "PowerPepper" name.

Trick #5: get lost in Windows shell command translation

The Windows shortcut (LNK) file from the LNK-based delivery chain actually starts a Windows shell with an obfuscated command argument. The command is a specific form of a “FOR” Windows shell loop that generates the “PowerShell” string from an “assoc” shell built-in result.



The malicious LNK file will fire a PowerShell script as a result, which in turn will recompose a second stage script from a downloaded Word document, as seen in Trick #1.

Trick #6: kick start it all with a signed binary proxy execution

Whether it’s at the end of macros execution (for the macro-based delivery chain) or as a last step of the shortcut-embedded scripts (for the LNK-based delivery chain), DeathStalker leveraged a signed binary proxy execution to start up PowerPepper for the first time.

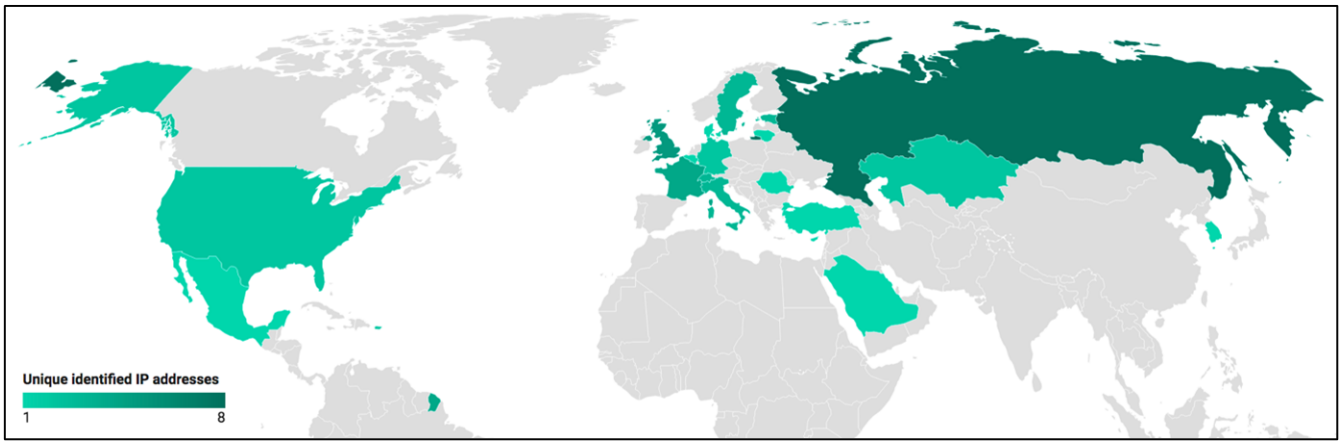
Visual Basic

```
1  $ttss=Join-Path -path $src -ChildPath ('Startup'+[char]92+'StartPrinter.url');
2  start-process -filepath 'rundll32.exe' -argumentlist
3  ('ieframe.dll,openurl '.replace('openurl',('o').toupper()+'pen'+('url').toupper())+$ttss)
```

While the first (macro-based) delivery chain we retrieved fired the malicious VBS loader with “rundll32.exe ieadvpack.dll, RegisterOCX wscript.exe <script file> <script argument> pexe”, more recent ones use a “rundll32.exe ieframe.dll, OpenURL <Internet shortcut>” alternative combo. The very latest rely on a dropped internet shortcut file (.url), which simply opens an LNK launcher with a “file://” URL. The LNK launcher in turn runs the VBS loader (see Trick #3).

Geography of PowerPepper's targets

We of course cannot get a comprehensive view of all PowerPepper's targets, but having tracked this implant since May 2020, we managed to get a partial view of targeted countries before August 2020, as well as in November 2020.



Due to the very partial information we sometimes get for such research, and despite our efforts to filter as much as we can, we cannot rule out that some identified targets could actually be fellow researchers investigating the threat, or DeathStalker's own testing infrastructure.

We could not precisely identify PowerPepper targets, but law and consultancy firms have been frequent targets of the actor.

Prevention and protection leads

In order to prevent successful PowerPepper execution or delivery, or to protect against related infection chains, we could not but underline these standard defense measures:

- **Content hosts** can regularly scan hosted files for malicious content, where regulations allow. They can protect their hosting infrastructure with endpoint protection software and traffic monitoring. They can also stack protection on privileged and remote access, with client network address filtering, multi-factor authentication (MFA), and auditing of authentication logs.
- **Website owners and editors** need to frequently and responsively update their CMS backends as well as associated plugins. They can also stack protection on privileged and remote access, with client network address filtering, MFA and access logging on all backend endpoints.
- **Enterprise IT services** need to restrict script engine (i.e., PowerShell) use on end-user computers with enforced execution policies. They need to set up endpoint protection software on end-user computers and content servers. They should allow DNS requests to corporate-managed resolvers and relays only, while filtering HTTP and DNS traffic at the perimeter. Last but not least, they need to train employees not to open attachments and links in emails from unknown senders.
- **Individuals** should never open Windows shortcuts that were downloaded from a remote location or attached to an email, open attachments or click links in emails from unknown senders, or enable macros in documents from unverified sources.

Conclusion

It only seems fair to write that DeathStalker tried hard to develop evasive, creative and intricate tools with this PowerPepper implant and associated delivery chains. There is nothing particularly sophisticated about the techniques and tricks that are leveraged, yet the whole toolset has proved to be effective, is pretty well put together, and shows determined efforts to compromise various targets around the world.

This is consistent with previous knowledge of the DeathStalker actor, which has demonstrated continuous capabilities to compromise targets since 2012, and has been fast to develop new implants and toolchains. We discovered the PowerPepper implant in May 2020, and it has been improved or adapted regularly since then. At the same time, we also uncovered another previously unknown malware strain that we strongly believe is from the same actor, though we haven't identified any Powersing-related activity since our previous article on DeathStalker in August 2020.

The DeathStalker threat is definitely a cause for concern, with the victimology for its various malware strains showing that any corporation or individual in the world can be targeted by their malicious activities, provided someone has decided they are of interest and passed on the word to the threat actor. Luckily for defenders, DeathStalker has, until now, relied on a rather limited set of techniques to design its delivery chains, and implementing counter-measures is an attainable goal for most organizations.

Indicators of compromise

File hashes

IOC	Description
A4DD981606EA0497BF9995F3BC672951	Malicious Word document (macro-based delivery chain)
871D64D8330D956593545DFFF069194E	Malicious Word document (macro-based delivery chain)
81147EDFFAF63AE4068008C8235B34AF	Malicious Windows shortcut (LNK-based delivery chain)
DFC2486DE9E0339A1B38BB4B9144EA83	Malicious Word document (downloaded by LNK-based delivery chain)
74D7DF2505471EADEB1CCFC48A238AEC	Malicious CHM container
5019E29619469C74F2B826535C5A8BD8	Malicious CHM container
B4790E70B1297215E0875CFC2A56648E	Malicious CHM container
3A6099214F474C1501C110CE66033F3C	Malicious VBS Loader
07308FBC3D10FD476F1898ECF6762437	Malicious VBS Loader
1F77FBE4702F787A713D394B62D27B42	Malicious VBS Loader
6E99F6DA77B0620E89F6E88D91198C32	Malicious VBS Loader
5D04D246F3E5DA6A9347EC72494D5610	Malicious startup launcher LNK
BA7AE1C73A78D8DC4B3779BD6A151791	Malicious startup launcher LNK
1DC2B849A858BC479B1EF428491E0353	Malicious startup launcher LNK
9D4066C57C6E1602CE33F15DC7F3841B	PowerPepper steganography image (peppers)
6FF8A3D18A6EA930E87AC364379ECEC2	PowerPepper steganography image (peppers)
871D64D8330D956593545DFFF069194E	PowerPepper steganography image (peppers)
9CE299BBDD7FDBF9F30F8935C89D2877	PowerPepper steganography image (ferns)
34F086AE78C5319FB64BF1CAE8204D1B	PowerPepper steganography image (ferns)

File paths

IOC	Description
%PROGRAMDATA%\Support\licenseverification.vbs	Malicious VBS Loader

%PROGRAMDATA%\Support\licenseverify.vbs	Malicious VBS Loader
%PROGRAMDATA%\MyPrinter\NewFile.vbs	Malicious VBS Loader
%PROGRAMDATA%\Printers\NewFile.vbs	Malicious VBS Loader
%APPDATA %\Microsoft\Windows\Start Menu\Programs\Startup\System.lnk	Malicious startup launcher LNK
%PROGRAMDATA%\MyPrinter\Web.lnk	Malicious startup launcher LNK
%PROGRAMDATA%\Printers\Web.lnk	Malicious startup launcher LNK
%APPDATA%\Roaming\Microsoft\Windows\Start Menu\Programs\StartUp\StartPrinter.url	Malicious startup launcher URL

Domain and IPs

IOC	Description
allmedicalpro[.]com	PowerPepper C2 domain name
mediqhealthcare[.]com	PowerPepper C2 domain name
gofinancesolutions[.]com	PowerPepper C2 domain name
mailsigning.pythonanywhere[.]com	PowerPepper Signaling hostname (legitimate host and root domain)
mailsignature.pythonanywhere[.]com	PowerPepper Signaling hostname (legitimate host and root domain)
mailservice.pythonanywhere[.]com	PowerPepper Signaling hostname (legitimate host and root domain)
mailservices.pythonanywhere[.]com	PowerPepper Signaling hostname (legitimate host and root domain)
footersig.pythonanywhere[.]com	PowerPepper Signaling hostname (legitimate host and root domain)
globalsignature.pythonanywhere[.]com	PowerPepper Signaling hostname (legitimate host and root domain)

URLs

IOC	Description
hxxps://www.gsn-nettoyage[.]com/wp-snapshots/btoken.php	PowerPepper Signaling hostname (legitimate but compromised website)
hxxps://www.gsn-nettoyage[.]com/wp-snapshots/etoken.php hxxps://www.gsn-nettoyage[.]com/wp-snapshots/1.docx hxxps://www.gsn-nettoyage[.]com/wp-snapshots/Quote 16 db room.docx	Malicious documents download location (legitimate but compromised website)
hxxps://outlookusers.page[.]link/	Malicious documents download location (legitimate host and root domain)
hxxps://1drv[.]ws/w/s!AvXRHBXCKmvYdifkocKujNavvjY?e=hhuBV8	Malicious document remote location (legitimate host and root domain)
hxxps://1drv[.]ws/w/s!AvXRHBXCKmvYdcbz1YwTJRkOxP4?e=u5wtbX	Malicious document remote location (legitimate host and root domain)

hxxps://1drv[.]ws/w/s!AvXRHBXCKmvYd1921tVEMKWaCUs?e=MyoVNF	Malicious document remote location (legitimate host and root domain)
hxxps://1drv[.]ws /w/s!AvXRHBXCKmvYeFdjVtZN0Quljs4?e=dnA6GG	Malicious document remote location (legitimate host and root domain)
hxxps://1drv[.]ws/w/s!AvXRHBXCKmvYeePNerfsAWK0qVY?e=e4SsYM	Malicious document remote location (legitimate host and root domain)
hxxps://1drv[.]ws/w/s!AvXRHBXCKmvYejBpdekg1WUCM9M?e=UkhU10	Malicious document remote location (legitimate host and root domain)
hxxps://1drv[.]ws/w/s!AvXRHBXCKmvYe1ulhtazjNVvCqY?e=WptVTC	Malicious document remote location (legitimate host and root domain)

Mail addresses

IOC	Description
a.christy_inbox@outlook[.]com	Suspected malicious spear-phishing email sender (legitimate root domain)