SPEAR PHISHING TARGETING ICS SUPPLY CHAIN – ANALYSIS

January 20, 2021

MARKEL PICADO

Table of Contents

Introduction

This report covers an attack investigation done by DeNexus Threat Intelligence targeting supply-chain companies in the Middle East. Threat Intelligence is one of three major data sets for risk modeling in our DeRISK platform. Using information about threats, tactics techniques and procedures (TTPs), indicators of compromise (IoCs), attacker's behavior patterns etc., DeRISK changes risk quantification for affected companies.

In September, 2020 ZScaler has published a report on a targeted attack on Oil and Gas Supply Chain Industries in Middle East. DeNexus Threat Intelligence has discovered additional details of this attack and new victims of this threat actor. The campaigns we have observed have evolved overtime, and the threat actor is still active with more campaigns.

In this report we explain these campaigns and the strategy the threat actor uses to infect targets.

Threat Analysis

This threat actor uses spear phishing for the initial access. We have detected at least 3 campaigns in the wild using the same or similar spear phishing email templates. To deliver the payload said threat actor uses different tactics, which are explained in the "Distribution strategy" section.

Some of these malicious emails have 0 detections in Virus Total (VT).

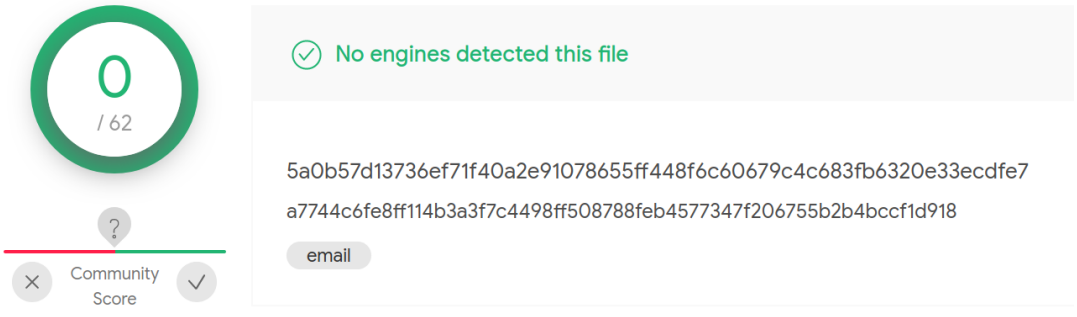


Image1: Email sample with 0 detections in VT

We currently know that this actor is trying to collect information from the companies that are under attack and mostly uses information stealers as final payloads. In most cases the information stealer is AgentTesla.

This kind of malware can be used in the early stages of an attack against a company and its main goal is to obtain credentials from employees to gain first access to the corporate network, which can lead to a more sophisticated attack.

Distribution Strategy

During different campaigns, we observed that the attacker has used similar distribution strategy, such as at the beginning the actor is using an email with a file attachment. This attachment usually is a RAR file, ZIP file or IMG file. The file contains the final payload with an information stealer.

In the most recent campaigns, the attacker uses an email with a PDF attachment. The PDF contains a link to a ZIP file hosted on a server controlled by the attacker or in a third-party file hosting like “mega.nz”. Inside the ZIP file there is a payload. Like in previous campaigns these payloads are information stealers.

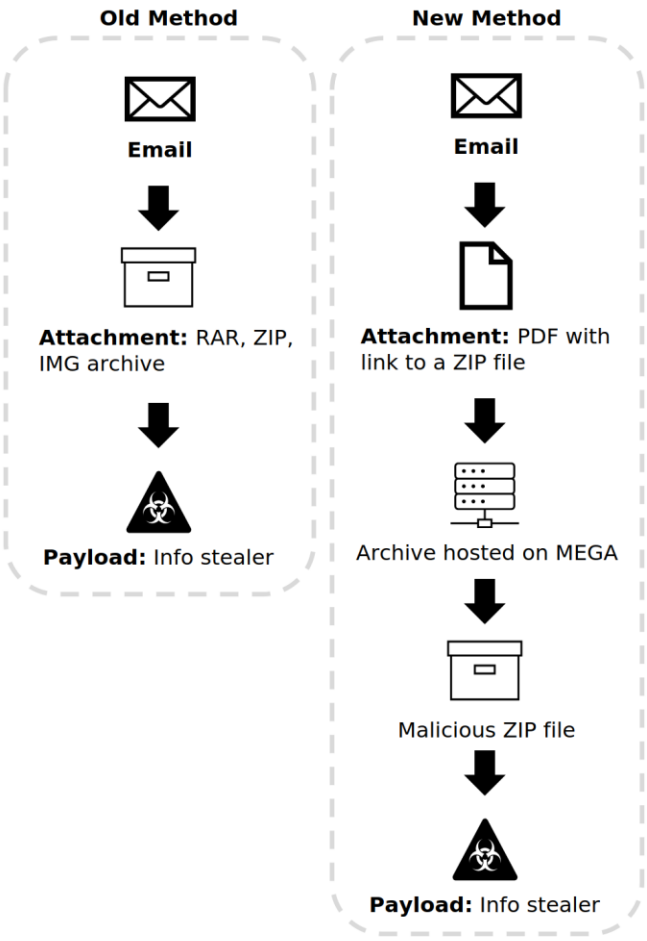
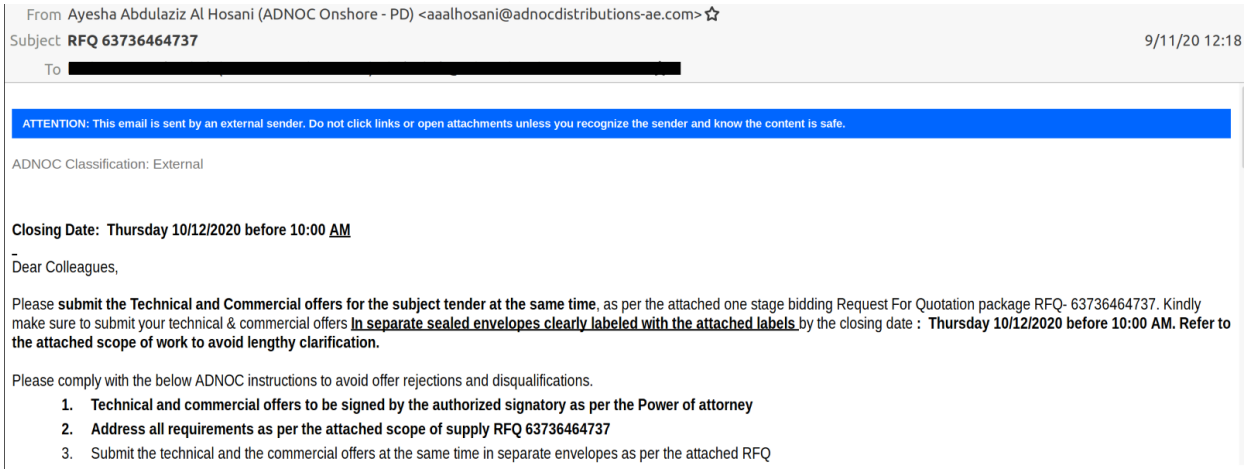


Image2: Payload delivery methods.

Spear Phishing

The emails we examined during all the campaigns had the same pattern; they were impersonating a legitimate company. In these emails, the attacker makes a request that must be managed before the end of the day, which creates a sense of urgency for the victim to open the malicious attachment. An example of this type of email can be found in Image 3.


From Ayesha Abdulaziz Al Hosani (ADNOC Onshore - PD) <aaalhosani@adnocdistributions-ae.com>
Subject **RFQ 63736464737** 9/11/20 12:18
To [redacted]

ATTENTION: This email is sent by an external sender. Do not click links or open attachments unless you recognize the sender and know the content is safe.

ADNOC Classification: External

Closing Date: Thursday 10/12/2020 before 10:00 AM

Dear Colleagues,

Please **submit the Technical and Commercial offers for the subject tender at the same time**, as per the attached one stage bidding Request For Quotation package RFQ- 63736464737. Kindly make sure to submit your technical & commercial offers **In separate sealed envelopes clearly labeled with the attached labels** by the closing date **: Thursday 10/12/2020 before 10:00 AM. Refer to the attached scope of work to avoid lengthy clarification.**

Please comply with the below ADNOC instructions to avoid offer rejections and disqualifications.

1. **Technical and commercial offers to be signed by the authorized signatory as per the Power of attorney**
2. **Address all requirements as per the attached scope of supply RFQ 63736464737**
3. Submit the technical and the commercial offers at the same time in separate envelopes as per the attached RFQ

Image3: Spear phishing example.

Not only the distribution methods have evolved, but also, the spear phishing campaigns have become more sophisticated. In the first campaigns, the threat actor used a random sender account or the real sender mail address, which didn't look like a real account from the targeted company.

As shown in the example above, the threat actor now registers a domain like the original victim's domain in order to make the email look even more real.

Identity Theft

In a recent campaign in November, we discovered that the threat actors are not only trying to pose as the targeted company, but also another company – steel production company from India, which is demonstrated in Image 4.

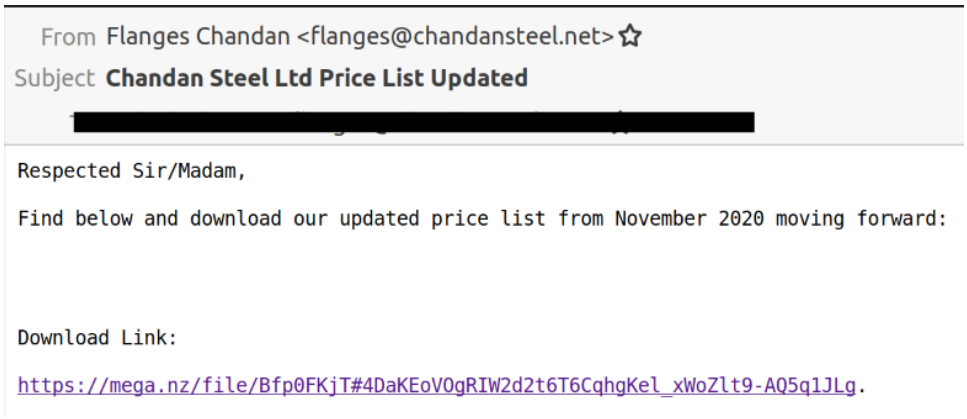


Image4: Spear phishing

In this case, the payload was also the same that was used in previous campaigns and it communicates with the same command and control center.

Image 4 above shows in this case the threat actor doesn't use attachments, they directly put the link to the malicious ZIP file in the body of the email.

As explained before, the most recent campaigns were using an attached PDF. In the previous example, the PDF appears to be a Requests for Quotations (RFQ) for supply contracts and legal tenders for various projects related to target company. The threat actor appears to have carefully crafted the PDF to make it appear legitimate, as seen in Image 5.

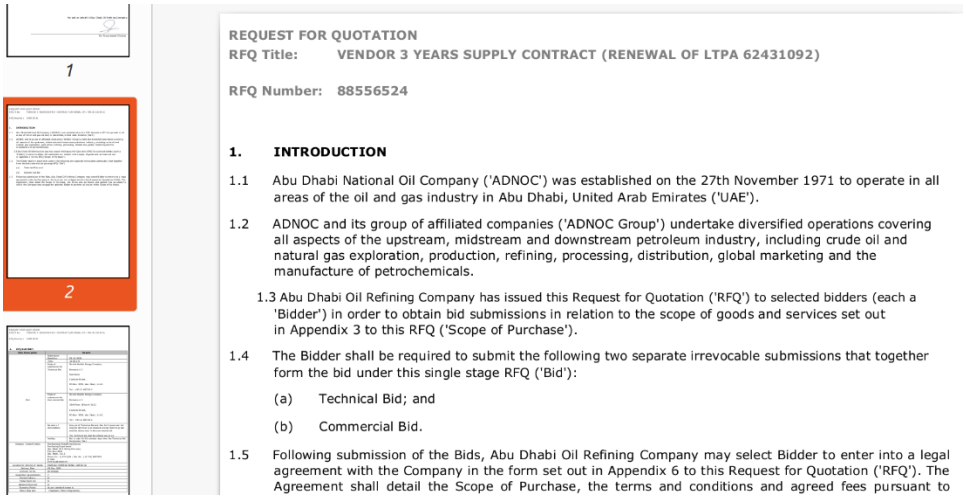


Image5: Crafted PDF.

The PDF at the time of analysis was not detected by any AV engine in VT, like the previous email which contains this PDF file as attachment.

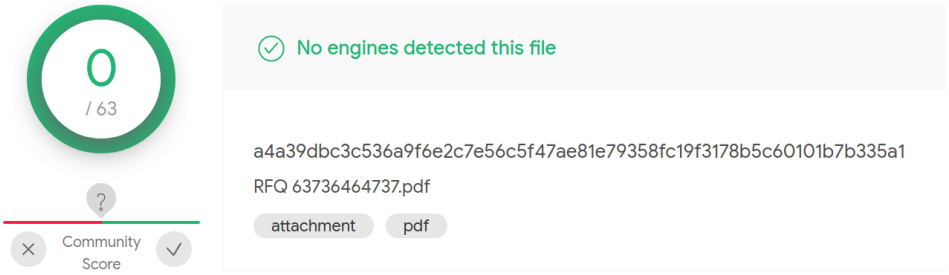
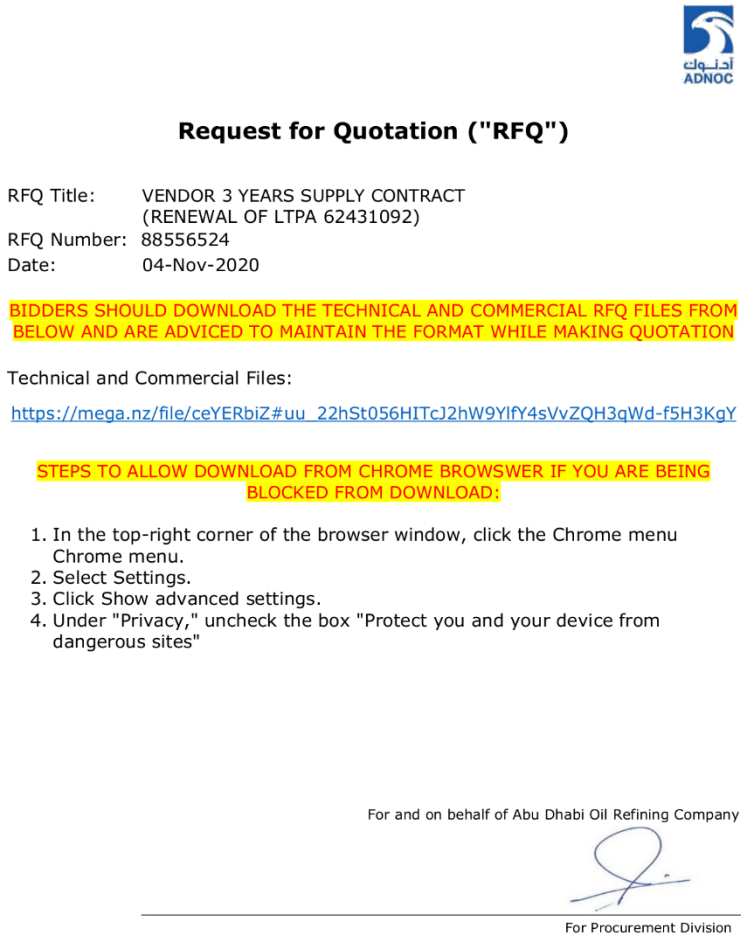


Image6: Crafted PDF with 0 detections in VT (It contains malicious URL).

Fortunately, the PDF does not exploit any vulnerability, nor does it execute code remotely. However, on the first page, once the victim opens the PDF, the reader will see a URL which points to MEGA, and steps to correctly download that file (even if the download is blocked by the browser). Image 7 shows this in more detail.



ADNOC

Request for Quotation ("RFQ")

RFQ Title: VENDOR 3 YEARS SUPPLY CONTRACT (RENEWAL OF LTPA 62431092)
RFQ Number: 88556524
Date: 04-Nov-2020

BIDDERS SHOULD DOWNLOAD THE TECHNICAL AND COMMERCIAL RFQ FILES FROM BELOW AND ARE ADVICED TO MAINTAIN THE FORMAT WHILE MAKING QUOTATION

Technical and Commercial Files:

https://mega.nz/file/ceYERbiZ#uu_22hSt056HITcJ2hW9YlfY4sVvZQH3qWd-f5H3KgY

STEPS TO ALLOW DOWNLOAD FROM CHROME BROWSWER IF YOU ARE BEING BLOCKED FROM DOWNLOAD:

1. In the top-right corner of the browser window, click the Chrome menu Chrome menu.
2. Select Settings.
3. Click Show advanced settings.
4. Under "Privacy," uncheck the box "Protect you and your device from dangerous sites"

For and on behalf of Abu Dhabi Oil Refining Company

For Procurement Division

Image7: First page of the crafted PDF. Containing malicious URL and instructions in case the browser doesn't allow to download the payload.

The victim will need to click on that link to download the payload. We believe that since the payload is not automatically downloaded and stored in MEGA, the AV companies are not classifying this file as malicious and could be the reason why the threat actor has changed the delivery method.

Once the victim clicks on the URL they find the following, as seen in Image 8:

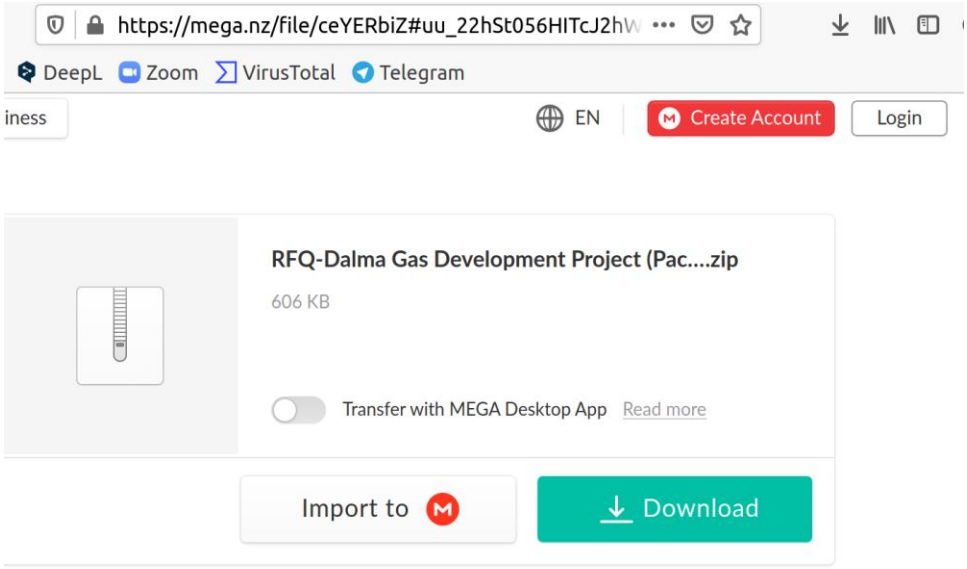


Image8: MEGA hosting the malicious payload.

The name of the file is “RFQ-Dalma Gas Development Project (Package B) -TENDER BULLETIN-13.zip”

Toolkit

The downloaded zip contains an executable file, which was the final payload.

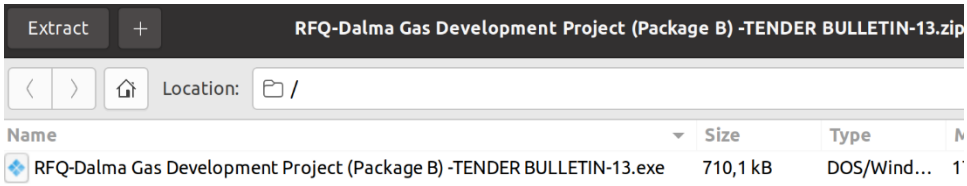


Image9: Content of the ZIP file hosted in MEGA.

In this case the payload was detected by 56 AV engines in VT, as shown in Image 10.

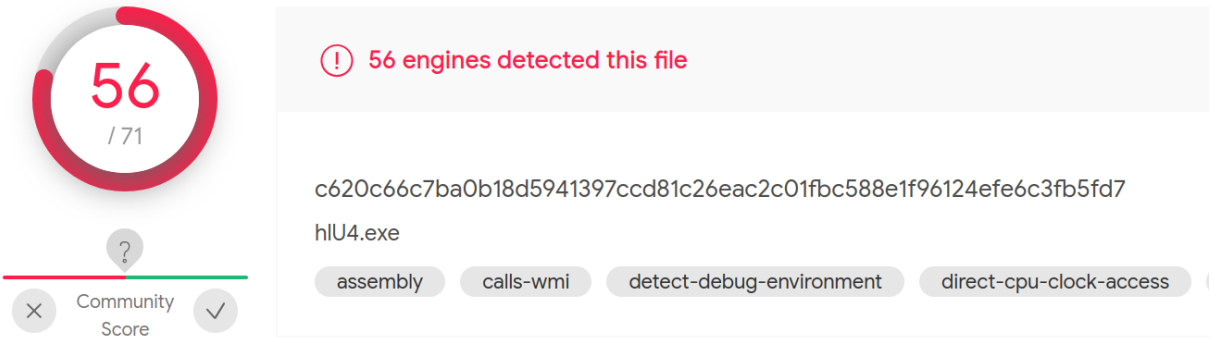


Image10: 56 detections in VT for the executable in the ZIP file. (packed Agent Tesla)

These actors have a variety of payloads, all of them are mainly information stealers. In this case, the payload is an AgentTesla, but during the investigation we found the following information stealers:

- AgentTesla (the most used)
- Formbook
- Masslogger
- Matiex
- AZORult

Since the malware we have mostly encountered was AgentTesla, the following section is an in-analysis in depth of this malware.

AgentTesla v3 Analysis

AgentTesla is a .NET-based spyware, keylogger, and information stealer that has the capability to steal data from different applications (Browsers, FTP Clients...). AgentTesla has been observed since 2014, and is still active. Throughout the years, new versions of this malware have appeared, to which more capabilities and improvements have been added.

Once the email is received, attachment is opened, and the executable hosted in MEGA is downloaded, the victim gets infected (packed AgentTesla).

File name	RFQ-Dalma Gas Development Project (Package B) -TENDER BULLETIN-13.exe
File sha256	c620c66c7ba0b18d5941397ccd81c26eac2c01fbc588e1f96124efe6c3fb5fd7

Image11: Payload

This executable is written using .NET framework. The sample is packed, and this packer has different stages in order to unpack the final payload (AgentTesla). In one of these stages, it creates a child process where the final payload is injected, (using process hollowing technique) this behavior is not related to AgentTesla but to its packer.

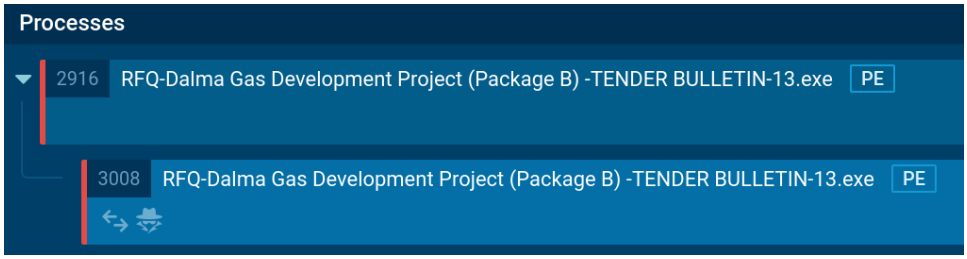


Image12: Process tree of the payload execution (Process Hollowing).

After the unpacking stages, the AgentTesla starts collecting credentials from the infected host and sending them to the attacker's server. In this case a SMTP server.

Packer

The packer of this AgentTesla is complex and highly obfuscated, containing different elements (DLLs) to unpack the final payload.

There are different stages which consist of:

1. Decrypt the different components (DLLs) which help on tasks like:
    1.1. Decrypt the final payload (AgentTesla)
    1.2. Inject the payload into a child process
2. Anti-debugging and anti-analysis techniques
3. Process injection

Not only obfuscation is implemented to make the analysis more difficult, but also there are some anti-debugging tricks implemented that crash the decompilers.

```
array[9] = (array[9] ^ array2[9]);
array[10] = array[10] * array2[10];
array[11] = array[11] + array2[11];
array[12] = (array[12] ^ array2[12]);
array[13] = array[13] * array2[13];
array[14] = array[14] + array2[14];
array[15] = (array[15] ^ array2[15]);
uint num12 = 64U;
u的fcN希z的Ipx.\u200E\u206E\u206C\u202C\u200B\u202B\u206C\u206B\u200D\u206A\u202D\u202C
  \u202D\u200B\u202A\u200E\u206C\u206A\u206C\u206E\u206E\u200B\u206A\u206A\u206A\u200F
  \u200F\u200B\u202D\u206A\u202B\u206C\u202E\u200D\u200E\u202A\u200D\u200F\u206E\u206A
  \u202E((IntPtr)((void*)ptr3), num3 << 2, num12, ref num12);
if (num12 == 64U)
{
    return;
}
uint num13 = 0U;
for (uint num14 = 0U; num14 < num3; num14 += 1U)
{
    *ptr3 ^= array[(int)(num13 & 15U)];
    array[(int)(num13 & 15U)] = (array[(int)(num13 & 15U)] ^ *(ptr3++)) + 1035675673U;
    u的fcN希z的Ipx.\u206D\u206F\u200D\u206B\u202E\u202D\u200B\u206C\u206E\u206F\u202C
      \u206E\u206E\u206D\u202E\u202D\u200C\u202B\u200E\u206F\u206F\u202B\u200D\u200B
      \u206F\u202A\u200F\u200C\u202B\u202A\u200F\u206B\u200F\u200E\u200E\u206D\u202D
      \u202A\u200F\u206B\u202E(Process.GetCurrentProcess().Handle, ref flag2);
    if (flag2)
    {
        Environment.FailFast(null);
    }
    num13 += 1U;
}
```

Image13: Packer's obfuscated code.

```
// 行商xU司fyf的l.u的fcN希z的Ipx
// Token: 0x0600000A RID: 10 RVA: 0x00004058 File Offset: 0x00002458
internal static - \u206B\u200B\u206B\u200E\u202C\u202C\u202B\u202E\u202C\u202A\u206D\u200E\u200F\u206C\u206E\u200C\u200C\u206E
  \u206B\u200E\u206E\u206D\u202C\u206B\u206B\u200D\u200F\u202D\u200D\u200C\u202D\u200F\u202A\u202C\u206A\u200C\u200E\u200B
  \u200C\u200D\u202E<->(uint)
{
    /*
An exception occurred when decompiling this method (0600000A)

ICSharpCode.Decompiler.DecompilerException: Error decompiling - 行商xU司fyf的l.u的fcN希z的Ipx::<->(System.UInt32)
 ---> System.NullReferenceException: Referencia a objeto no establecida como instancia de un objeto.
   en ICSharpCode.Decompiler.ILAst.ILInlining.CanPerformCopyPropagation(ILExpression expr, ILVariable copyVariable) en D:\a
     \dnSpy\dnSpy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\ILAst\ILInlining.cs:línea 561
   en ICSharpCode.Decompiler.ILAst.ILInlining.CopyPropagation(List`1 newList) en D:\a\dnSpy\dnSpy\Extensions\ILSpy.Decompiler
     \ICSharpCode.Decompiler\ICSharpCode.Decompiler\ILAst\ILInlining.cs:línea 504
   en ICSharpCode.Decompiler.ILAst.ILAstOptimizer.Optimize(DecompilerContext context, ILBlock method, AutoPropertyProvider
     autoPropertyProvider, StateMachineKind& stateMachineKind, MethodDef& inlinedMethod, AsyncMethodDebugInfo& asyncInfo,
     ILAstOptimizationStep abortBeforeStep) en D:\a\dnSpy\dnSpy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler
     \ICSharpCode.Decompiler\ILAst\ILAstOptimizer.cs:línea 234
   en ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(IEnumerable`1 parameters, MethodDebugInfoBuilder&
     builder) en D:\a\dnSpy\dnSpy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast
     \AstMethodBodyBuilder.cs:línea 123
   en ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(MethodDef methodDef, DecompilerContext context,
     AutoPropertyProvider autoPropertyProvider, IEnumerable`1 parameters, Boolean valueParameterIsKeyword, StringBuilder sb,
     MethodDebugInfoBuilder& stmtsBuilder) en D:\a\dnSpy\dnSpy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler
     \ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:línea 88
   --- Fin del seguimiento de la pila de la excepción interna ---
   en ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(MethodDef methodDef, DecompilerContext context,
     AutoPropertyProvider autoPropertyProvider, IEnumerable`1 parameters, Boolean valueParameterIsKeyword, StringBuilder sb,
     MethodDebugInfoBuilder& stmtsBuilder) en D:\a\dnSpy\dnSpy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler
     \ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:línea 92
   en ICSharpCode.Decompiler.Ast.AstBuilder.<>c__DisplayClass90_0.<AddMethodBody>b__0() en D:\a\dnSpy\dnSpy\Extensions
     \ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstBuilder.cs:línea 1519
*/;
}
```

Image14: Error when decompiling obfuscated code (DNSpy).

The unpacked sample hash is the following:

e6c7c02a7019cde94b0788aba4163251220e971a357381fca94baccc3a14901f

Capabilities

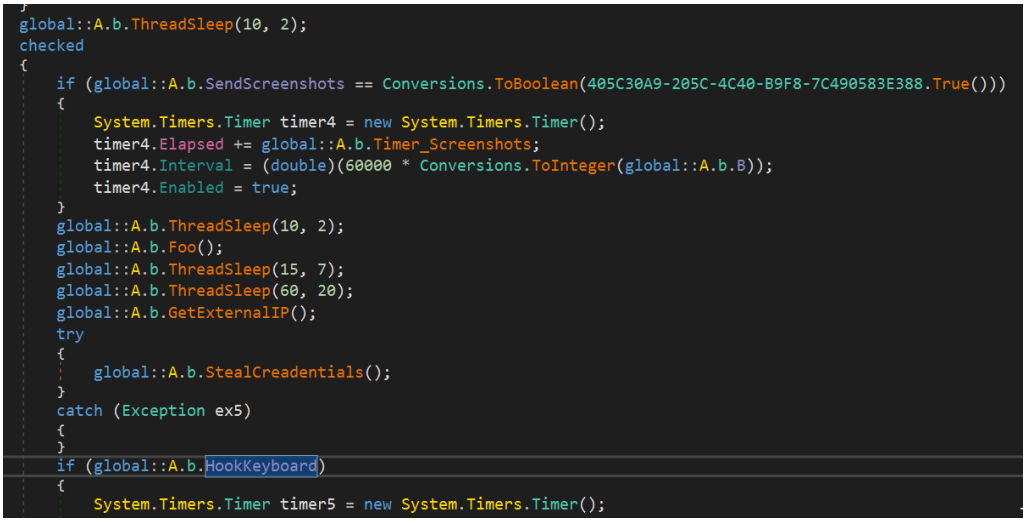
AgentTesla is an information stealer, and it has different features. All these features can be configured enabled/disabled from its configuration file.

Configuration includes:

- Persistence (enable/disable)
- Installation folder
- Registry key names for persistence
- To choose the network protocol to send stolen credentials
    - HTTP
        - Configure TOR proxy (enable/disable)
    - SMTP
    - FTP
    - Telegram

For this reason, the behavior of AgentTesla across different samples could be very different. In any case, the following list and Image 15 show AgentTesla features which could be enabled or disabled.

- Persistence (**configurable**)
- Screenshots (**configurable**)
- Hook keyboard (**configurable**)
- Network protocol used to send stolen credentials to C2 (**configurable**)
- Steal cookies
- Steal credentials
- Sandbox evasion technique
- Obfuscated code
- Strings Encryption
- System fingerprint



```
global::A.b.ThreadSleep(10, 2);
checked
{
    if (global::A.b.SendScreenshots == Conversions.ToBoolean(405C30A9-205C-4C40-B9F8-7C490583E388.True()))
    {
        System.Timers.Timer timer4 = new System.Timers.Timer();
        timer4.Elapsed += global::A.b.Timer_Screenshots;
        timer4.Interval = (double)(60000 * Conversions.ToInteger(global::A.b.B));
        timer4.Enabled = true;
    }
    global::A.b.ThreadSleep(10, 2);
    global::A.b.Foo();
    global::A.b.ThreadSleep(15, 7);
    global::A.b.ThreadSleep(60, 20);
    global::A.b.GetExternalIP();
    try
    {
        global::A.b.StealCreadentials();
    }
    catch (Exception ex5)
    {
    }
    if (global::A.b.HookKeyboard)
    {
        System.Timers.Timer timer5 = new System.Timers.Timer();
```

Image15: Hook Keyboard and Screenshots features.

For the analyzed sample, the configuration is the following:

- Persistence (**disabled**)
- Screenshots (**disabled**)
- Hook keyboard (**disabled**)
- Network protocol used to send stolen credentials to C2 (**SMTP of compromised company**)
- Steal cookies (**enabled**)
- Steal credentials (**enabled by default**)
- Sandbox evasion technique (**enabled by default**)
- Obfuscated code (**enabled by default**)
- Strings Encryption (**enabled by default**)
- Steal credentials (**enabled by default**)
- System fingerprint (**enabled by default**)

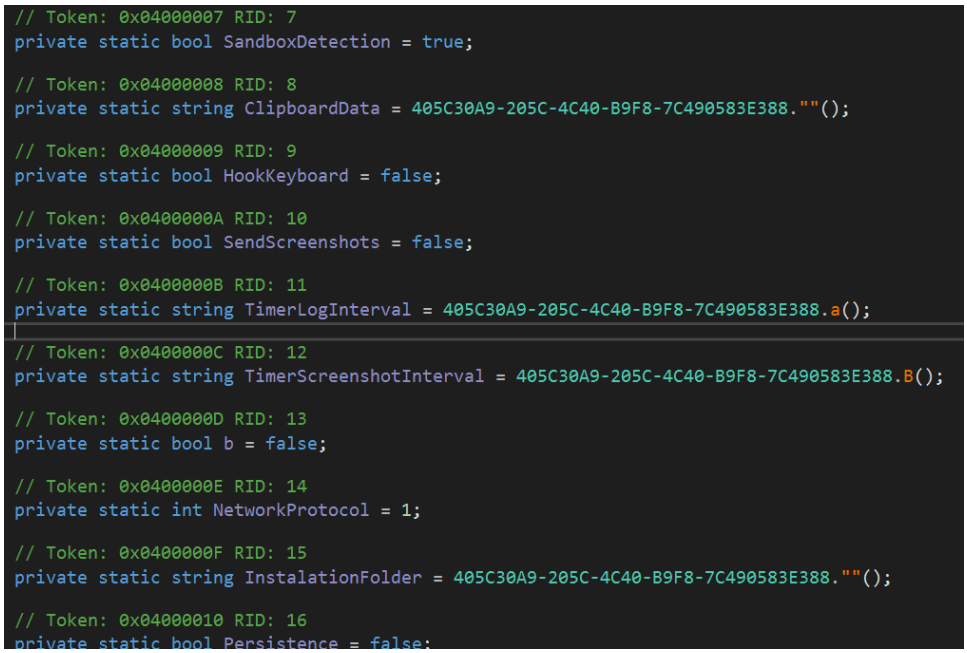


Image16: AgentTesla configuration.

Command & Control

AgentTesla has capabilities to send data using 4 different protocols:

- HTTP
- SMTP
- FTP
- Telegram

The protocol type used in each sample depends on its configuration. Samples that we've observed in our research are using SMTP commutation protocol. The usual setup to achieve this configuration is illustrated in Image 17.

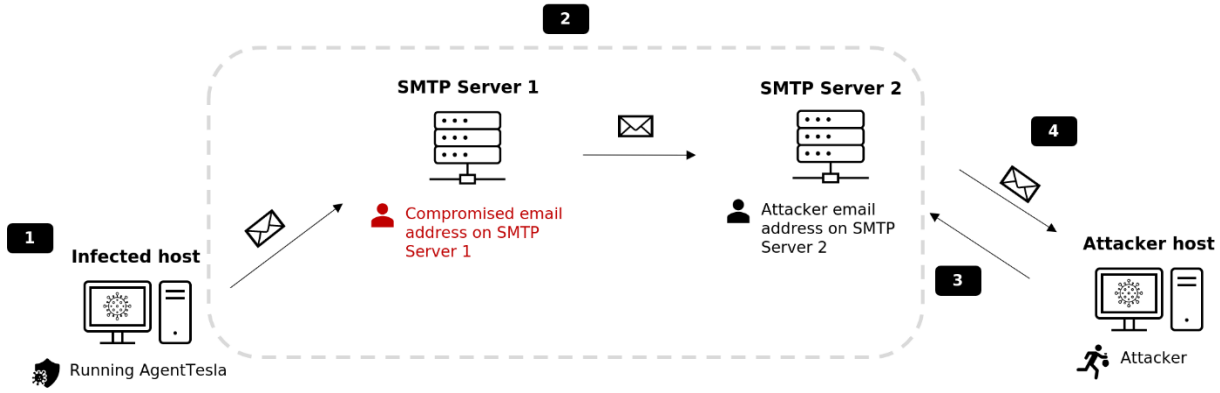


Image17: AgentTesla infastructure setup.

1. AgentTesla collects credentials using cookies from different applications on the infected host.
2. The infected data is sent to the attacker email address through a third-party SMTP server. Usually, the attacker has credentials of a compromised account on that SMTP server and AgentTesla is using these credentials to send stolen data.
3. The attacker connects to the SMTP servers.
4. Attacker extracts all stolen credentials from the infected host.

The setup is a bit different in this sample as you can see in Image 18.

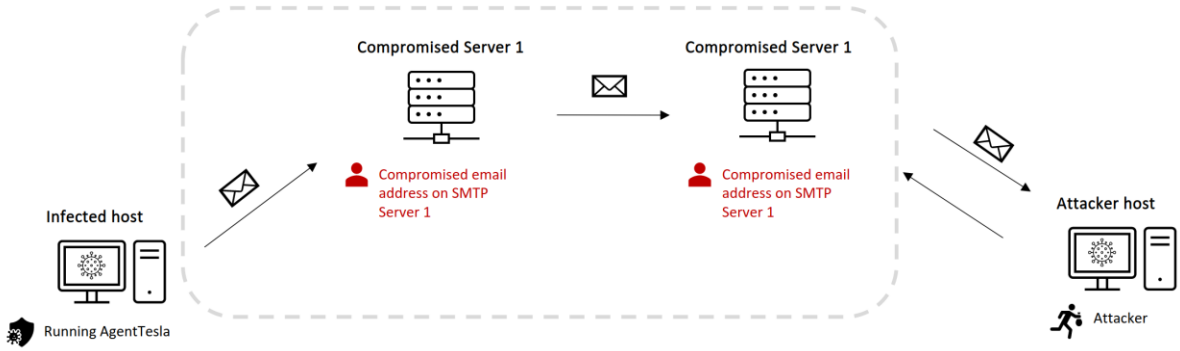


Image18: AgentTesla infrastructure setup for analyzed sample.

In this case the attacker is using the same compromised account for both sender and recipient.

Gathering data (Leaked credentials)

As we mentioned before, AgentTesla is an information stealer and has the capabilities to steal credentials from the following different applications:

- Browsers

- FTP clients
- VPN clients
- Mail clients

Depending on which AgentTesla version is used, it can steal information from different applications.

The following table shows some of these application names from which AgentTesla can get credentials.

Chrome	Safari	Outlook	FileZilla
IExplorer	SeaMonkey	Thunderbird	WinSCP
Firefox	Flock	PocoMail	SmartFTP
Opera	Comodo	Eudora	Pidgin
Yandex	Chromium	FoxMail	FlashFXP

Attack Surface

This section explains how the attacker operates based on the collected information from our active monitoring of the threat.

The attack is done as follows:

1. **Spear phishing** (Targeting ICS companies).
    a. In emails they send, they **pose as legitimate companies** (ADNOC in this case)
2. **Payload delivery.**
    a. Attachments
        i. **Archives** containing the payload (an PE executable)
            1. RAR files
            2. IMG files
            3. ZIP files
        ii. **Pdf files** containing a link to mega.mz
3. **Information Gathering**
    a. Toolkit
        i. Agent Tesla (the most used)
        ii. Formbook
        iii. Masslogger
        iv. Matiex
4. **Exfiltration**
    a. Sending stolen data back to a server controlled by the actor

On a successful attack, the threat actor can collect information from internal network servers, mail addresses and different resources which could lead on to a second stage of the attack. This time targeting internal enterprise network, and continue with internal spear phishing, which could lead to lateral movements across internal network.

MITRE ATT&CK Mapping

Image 19 shows the threat actor operations mapped to MITRE ATT&CK framework, which is also outlined in the chart below.

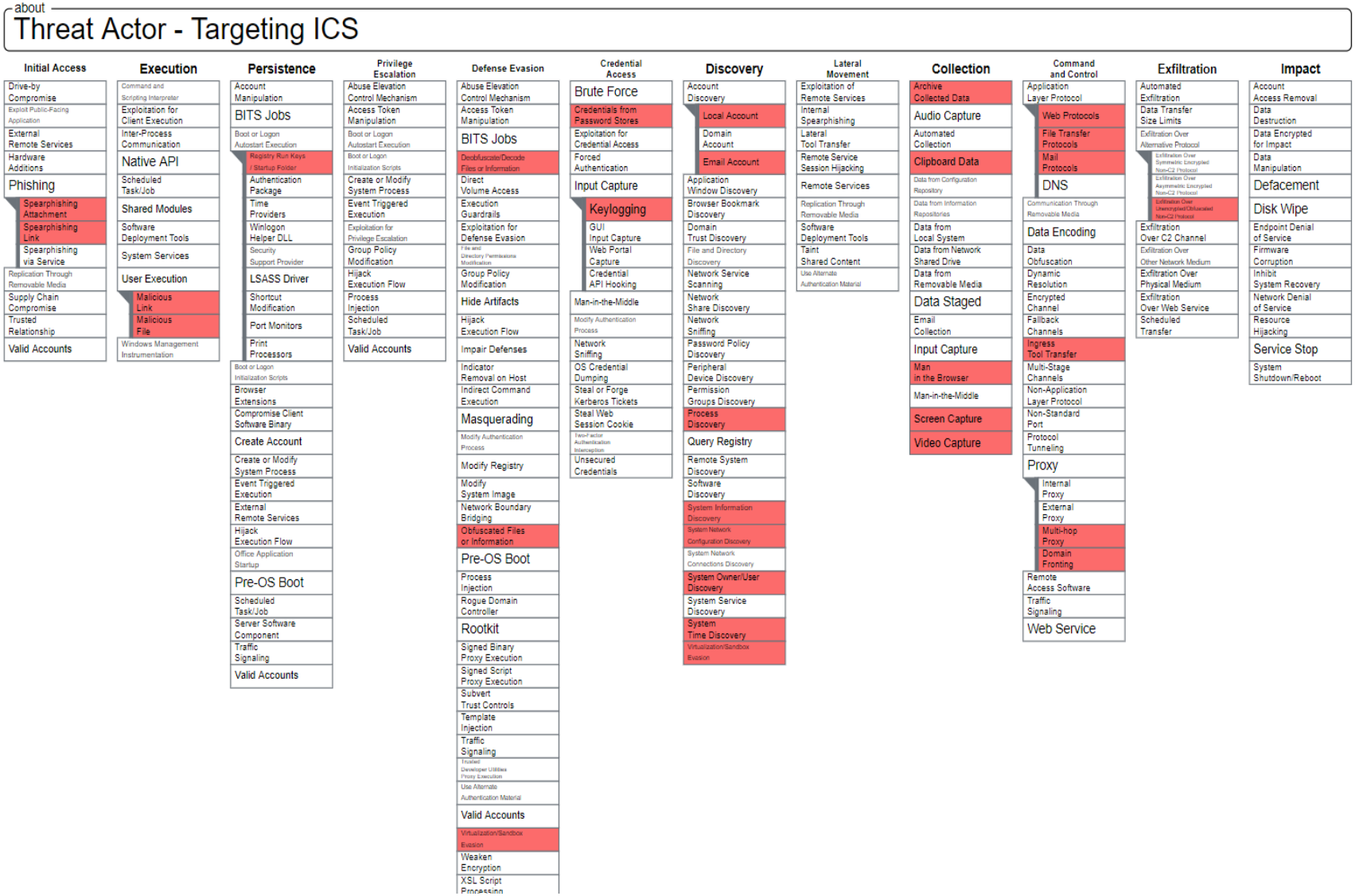


Image19: Threat actor's MITRE ATT&CK Mapping.

ID	Tactic
T1566.001	Spearphishing Attachment
T1566.002	Spearphishing Link
T1566.003	Spearphishing via Service
T1204.002	Malicious File
T1204.001	Malicious Link
T1547.001	Registry Run Keys / Startup Folder
T1140	Deobfuscate/Decode Files or Information
T1027	Obfuscated Files or Information
T1497	Virtualization/Sandbox Evasion
T1555	Credentials from Password Stores
T1056.001	Keylogging
T1057	Process Discovery
T1082	System Information Discovery
T1016	System Network Configuration Discovery
T1033	System Owner/User Discovery
T1124	System Time Discovery
T1497	Virtualization/Sandbox Evasion
T1087.003	Email Account
T1087.001	Local Account
T1560	Archive Collected Data
T1115	Clipboard Data
T1185	Man in the Browser
T1113	Screen Capture
T1125	Video Capture
T1105	Ingress Tool Transfer
T1071.002	File Transfer Protocols
T1071.003	Mail Protocols
T1071.001	Web Protocols
T1090.003	Multi-hop Proxy
T1048.003	Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol

Threat Actor Infrastructure

Image 20 shows a part of the threat actor's infrastructure.

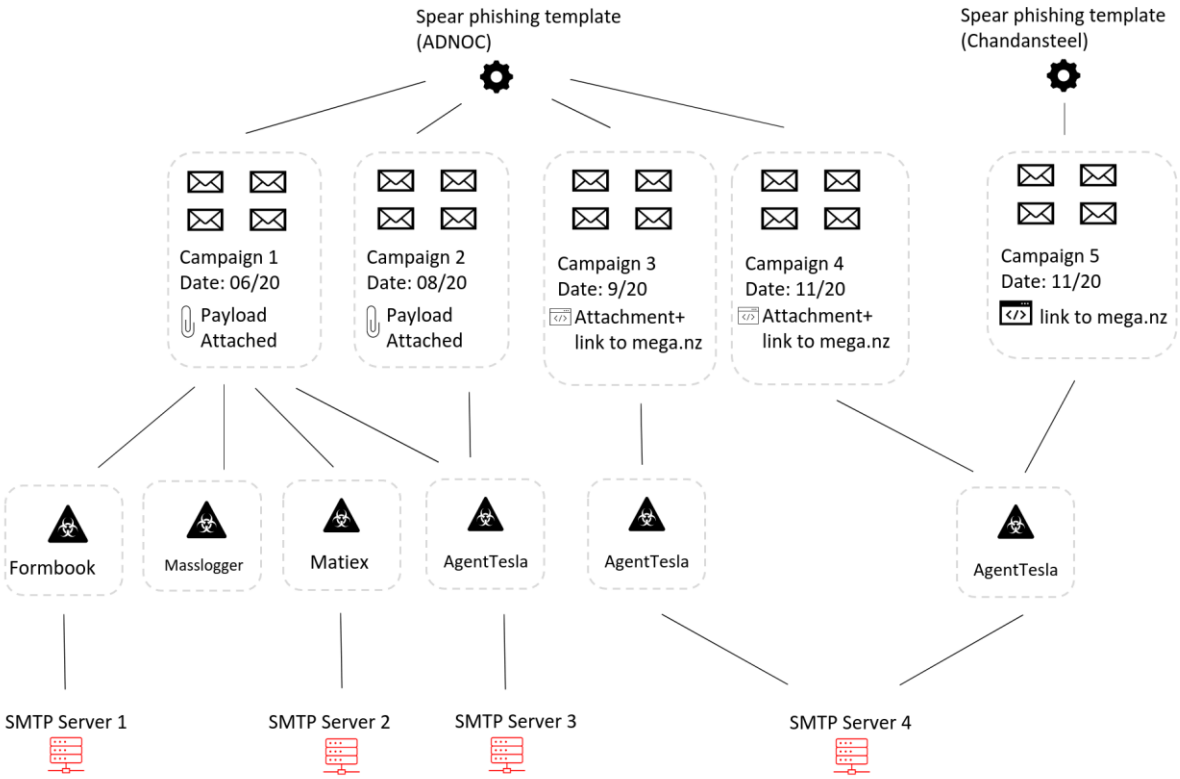


Image20: Threat actor's infrastructure.

Targeted Companies

As explained before, we have observed 3 different campaigns from the same threat actor during the 2020.

The targeted companies are listed below but it's not limited to these companies.

Location	Description
Europe	Commercial refrigerator supplier
Europe	Production of heavy electrotechnical equipment
APAC	Industrial Process & Factory Automation
Europe	The leading provider of smart automation solutions for functional safety
Middle East	International Maritime Industries

APAC	Produces diverse products from exterior and interior construction materials, paints and new materials
Europe	In operation in over 600 plants in more than 60 countries worldwide
North America	Custom manufacturing anti-slip covers in the USA
APAC	Providing the service which is most various on the transportation route which is most various
Europe	Production of optical components

Compromised Companies

After the attempts to recover the stolen information from different sources, we have detected that the following companies have been infected by this threat actor.

This means that the first stage of this attack was successfully done, and this threat actor has got credentials to access different resources of the infected company.

This list shows only a few companies for which the attacker has managed to compromise their employees, this list is bigger, and it may increase over time.

Location	Description
India	The manufacture of critical aerospace systems and high-value geospatial services
South Korea	Development company headquartered in Seongnam, South Korea that operates the search engine
UAE	Offers technical and engineering services to the upstream and downstream oil and gas industry of the middle east countries
China	Chinese oil and gas company and is the listed arm of state-owned
China	Owns one of the world's largest PTA plants, one of the world's largest functional fiber production bases and weaving enterprises
UAE	The company currently has three companies in this sector providing logistics contract services, freight forwarding , warehousing, and transportation

Sri Lanka	The national oil and gas company of Sri Lanka

Conclusion and Final Thoughts

Based on the type of malware used by the attacker and the companies it attacks, one could almost certainly say that this actor is mainly targeting employees in the supply chain industries in Oil and Gas sector in the Middle East. Besides Middle East, this attacker has also shown some interest in other regions (Europe, North America, APAC).

They use social engineering to trick employees with something that may be of interest to them, creating a sense of urgency and posing as a legitimate UAE based company or steel production company from India.

The fact that they try to impersonate different companies leads us to believe that for UAE-based targets, they are impersonating UAE companies, and for targets based in India they pose as Indian companies. This level of detail makes us think that the actor takes his/her time to craft carefully targeted emails.

The attacker has managed to compromise some of the employees of his/her target companies, and this could be used in future attacks. Due to the type of malware they use, it does not look like a nationally sponsored actor, so the actor's intentions may be related to money.

Because of the type of malware they use, it does not look like a nationally sponsored actor, so the actor's intentions may be related to money.

DeNexus's Threat Intelligence team will continue this investigation in order to gather more information about victims, attacker's tactics changes and other important information that is used to feed risk modeling and quantification mechanisms and to inform our customers.

IOCs

Emails
11ed9bf7c7542a32e5cbd683c4c33017
512ac83de44c841142960c13a95fa007
aa279b552eaed25e84147e1aa1cc4753
b327eb72ee2350d6d157fa408bbbf6b6
c31f2cbbcc4935d93d849ae7a018086a
1ad0afc08c2deb6d9b5ea25df68c8ce9
3789aba8ddbdc5889fdd66b1bf541fbf
20d8c7a9e913a3280cf37928ba565f44
d3fb9c680dd73386072da458dcead259

f63102d1b5e785adf639c7dfd84e1e4f
42fb2884621840a85b4c27e15653d7d1
16ab4b15d979890eb7e8fb948e49d7d6
be20a1f0fd5a784362162039bf3f060d
431eec759397c0eeeadd00c35f93d3c8
3bda4a12b5e107de88341133c336dab1
e9fea7adbb815b4ced8373fd80f963b5
bf59c73bf78622594cc1ae62db43bc40
835813f49403f3c5b27b7efffcfd8015
0a48604eec97cada00a552ccfef4a9f2
8df80532808e30def795d19252fc048f
bb9233de8542a6db7812f762630db78d
5f83a1acd943aabd65dbb1565954dd86
ef523d8ac425249bc852b1fb50a9f159

Payload
022149a1dd7335d6ac7772612fff400a
0802d930441ef6d1826411f2e0835c44
10216fd1ab967e8b6ed08bb48bef5372
1c71cfeafea353b7129770c0ae8d01e9
1fe390890cb65be3636df9f9c8617636
21a70080611b3745bd23345e82d22ab3
30c35b72cfd2ab1fb7b250b2a2cd6712
38940fb63b4c6344b88e2d70f683ee7b
4bb6055cfcb5a80dbb580dcb66f8a87f
4c6fa8d37bae0b6b35c3f2dd5c2e63ba
5beb45dbf2d92a6a3dfa335bb6fb5ae6
61aec386f231e1e9294c7c74054b6af3
624f757c1cad238790fd1aba66dcb719
7b5f158b6d4db5498c55d236798ce13b
810900643a4dc42b6339e81b9d24680f
8612ece9e9e0e5bc387570f6b2de8d0a
8cf4dddc8a21d1c77797b151ce42f5a5
910d96979ef235aa3ab50c4ffdab83fd
9eda8430e6bf0bab3f1e7134b584cd1b
a052addc22d6b9f0b18194c62b3da6e6
a3b579279547583d324a78045f93c777
aa0db1557b05fa2401ef31ac45900b04
b39b909a2a682f1a3c30d459a84c9e57
b5a1381ddf45028d9bf4bf5da815a0a7
b6322dc0d126f37a842d68ed1fa63160
bad63aab389957a96e6134c44a31aa9f
d7d8068a6359a97402ee1ba679eed000
e059e5e78ec388b64d4d98eb9ac7dff6
e866eba00cd16e6469f389525528f093
f2d4cc7225dc56808d760355eb53e8ac
f952ba9e5168f9ec7b08e7740d394e47
fc8def184bf6d9858a493871b1f4c53a
feaf8ab1cde85c530a0444b8fe2ce3f7