CERT-EU
FOR THE EU INSTITUTIONS, BODIES AND AGENCIES

Security Advisory 2019-002

Privilege Escalation Exploiting MS Exchange

January 31, 2019 — v1.0

TLP:WHITE

History:

- *31/01/2019 — v1.0 – Initial publication*

Summary

A vulnerability was discovered in Microsoft Exchange Server that allows a regular user to perform a privilege escalation technique and gain Domain Administrator access. Abusing the privileged role Exchange servers normally have by default on Active Directory (AD) infrastructures, an attacker can impersonate a mail server, perform an NTLM relay attack, and gain access to the Domain Controllers secrets (e.g. NTLM hashes and Kerberos keys).

Technical Details

This NTLM replay attack has been out in the wild for some time. In short, it allows privilege escalation. It is most commonly used over SMB, but it is also available over HTTP. More details can be found in an article written by Dirk-Jan Mollema [1]. The article explains how the attack can be performed using valid user credentials, or even without them. In the latter case, the attacker replays the authentication of an existing user in the same network segment as the Exchange server.

An attacker with access to a Microsoft Exchange mailbox can ask the considered server to send push notifications to an arbitrary URL that he or she controls. When sending an HTTP notification message to the requested location, Microsoft Exchange will authenticate itself by providing the server's NTLM hash. Using the latter, the attacker will be able to run an NTLM relay attack against one of the Domain Controllers, impersonating the Exchange Server. Abusing the privileged role the server normally has by default in Active Directory infrastructures (i.e. `WriteDacl` on the AD Domain object), the attacker will be able to grant the initial unprivileged user DC-Sync privileges (i.e. `Replicating Directory Changes` and `Replicating Directory Changes All`) and so the right to collect all the available NTLM hashes and Kerberos keys.

The tools `PrivExchange` [2] and `impacket` [3] that can be used for proof of concept are available on GitHub.

Versions Affected

The vulnerability has been proven in Microsoft Exchange 2013 (CU 21) on Windows Server 2012 R2 relayed to a Windows Server 2016 Domain Controller and on Microsoft Exchange 2016 (CU 11) on Windows Server 2016 relayed to a Windows Server 2019 Domain Controller (all fully patched) [1].

Recommendations

Mitigation

Having tested some of the mitigations proposed in [1], we have concluded that the following one works without any apparent adverse side-effects:

If Exchange Web Services (EWS) dynamic subscriptions [5] are not used in your organization, you can block the attack described in this advisory by **enforcing a limit of 0 subscriptions maximum on the EWS service across the whole Exchange organization**. This is done in the following way, according to [6]:

```
New-ThrottlingPolicy -Name AllUsersEWSPolicyNoSubscriptions -EwsMaxSubscriptions 0
  -ThrottlingPolicyScope Organization
```

If immediate enforcement is critical, you can run the following command on every front-end server; if not, you can skip this second step:

```
Restart-WebAppPool -Name MSExchangeServicesAppPool
```

Other Possible Mitigations

Additional mitigations are proposed in [1]. We have selected some and assessed their effectiveness. The results can be found below. For the sake of completeness, we also include some comments despite the fact they refer to solutions that have not been tested by us.

Tested

- **Enable Extended Protection for Authentication on the Exchange endpoints in IIS.**

This is about enabling Extended Protection Authentication [7] on at least one endpoint of the Exchange front-end sites found in Internet Information Services (IIS). According to our research and proof-of-concept, enabling it on the Exchange Web Services (EWS) endpoint is enough to break the attack. Following [8], two configurations need to be set up for this to work:

1. Create a Service Principal Name (SPN) for HTTP services running on the Exchange host machine;
2. Update the Web Services Virtual Directory pointing to the EWS endpoint with the `ExtendedProtectionTokenChecking` attribute set to *Require* and the `ExtendedProtectionSPNList` attribute set to the SPN created before.

```
setspn -s http/[Exchange Server FQDN]:[TCP Port] [Exchange Server NetBIOS]
Set-WebServicesVirtualDirectory -Identity "[Exchange Server NetBIOS]\EWS (Default Web Site)"
    -ExtendedProtectionTokenChecking Require -ExtendedProtectionSPNList http/[Exchange Server
      FQDN]:[TCP Port]
```

Comment: *Implementing this change has proven to break at least the results of the `Test-WebServicesConnectivity` cmdlet that performs a `GetFolder` test on the EWS Web Service endpoint. Implementing this change should be done with close monitoring of all Exchange services that are used.*

- **Remove the registry key which makes relaying back to the Exchange server possible, as discussed in Microsofts mitigation for CVE-2018-8518.**

Comment: *This does not seem to prevent the attack in our testing.*

Not Tested

- **Remove unnecessary privileges of Exchange to Domain**

Comment: *This recommendation is best targeted at organizations having access to Microsoft Support, as the special settings to setup Exchange with limited rights are not in the public domain and will very from version to version. Moreover, special care should be used to detect possible side effects on the various capabilities used in the Exchange architecture.*

- **Enable LDAP signing and channel binding.**

Comment: *Be aware of the possible side effects impacting non-Microsoft clients or older Microsoft operating systems in the ecosystem.*

- **Block connections on the Exchange level to arbitrary ports.**
- **Enforce SMB signing on Exchange servers (and preferable all other servers and workstations in the domain) to prevent cross-protocol relay attacks to SMB.**

References

[1] https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/

[2] https://github.com/dirkjanm/privexchange/

[3] https://github.com/SecureAuthCorp/impacket/

[4] https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2018-8581

[5] https://docs.microsoft.com/en-us/exchange/client-developer/exchange-web-services/notification-subscriptions-mailbox-events-and-ews-in-exchange

[6] https://docs.microsoft.com/en-us/powershell/module/exchange/server-health-and-performance/new-throttlingpolicy?view=exchange-ps

[7] https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-3.5/dd767318(v=vs.90)

[8] https://docs.microsoft.com/en-us/powershell/module/exchange/client-access-servers/set-webservicesvirtualdirectory?view=exchange-ps