# Microsoft leaks TLS private key for cloud ERP product

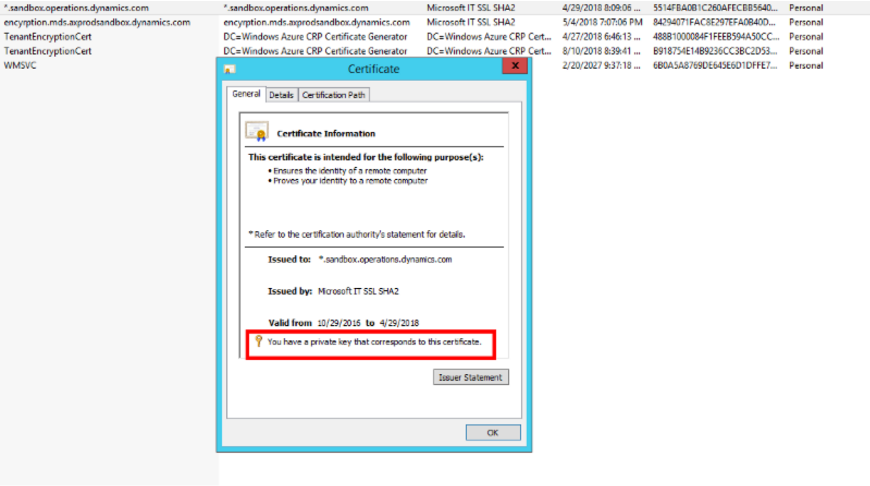
## … and it's still in use more than 100 days after the initial report

At my old job I was working as a software developer customising the Microsoft ERP product Dynamics 365 for Operations (formerly known as Dynamics AX). To provide some context: ERP stands for **E**nterprise **R**esource **P**lanning and is software which supports all critical core business processes of a company like purchasing, manufacturing, product planning, sales, finance and many others by integrating those into one application with a single database.

Last year Microsoft started to offer it's ERP product as a web-based cloud hosted SaaS solution. The software is hosted in Azure on a VM managed by Microsoft. It's accessible through a comprehensive control panel (Life Cycle Services) which empowers the user to manage every single aspect of the environment, like the deployment of changes to the software or applying updates.

One quite useful feature is direct RDP access to the machine running the software to debug issues with the application. A normal deployment is divided into at least three systems: development, user acceptance testing (also referred to as "sandbox") and production. The user acceptance system mirrors the exact setup of the production environment with a single exception: while there's no way to access the production servers besides through the web interface, the sandbox environment offers administrative RDP access. **And that's where the fun begins ;-)**

After a long work day while being off the clock I accessed a sandbox environment via RDP to take a look and learn how Microsoft would setup a server hosting such a business critical application. The hostname for a sandbox environment is *customername.sandbox.operations.dynamics.com*. A quick glance at the certificates inside the built-in "Certificate Manager" revealed something shocking:



| | | | | | |
|---|---|---|---|---|---|
| *.sandbox.operations.dynamics.com | *.sandbox.operations.dynamics.com | Microsoft IT SSL SHA2 | 4/29/2018 8:09:06 ... | 5514FBA0B1C260AFECBB5640... | Personal |
| encyrption.mds.axprodsandbox.dynamics.com | encyrption.mds.axprodsandbox.dynamics.com | Microsoft IT SSL SHA2 | 5/4/2018 7:07:06 PM | 84294071FAC8E297EFA0B40D... | Personal |
| TenantEncryptionCert | DC=Windows Azure CRP Certificate Generator | DC=Windows Azure CRP Cert... | 4/27/2018 6:46:13 ... | 488B1000084F1FEEB594A50CC... | Personal |
| TenantEncryptionCert | DC=Windows Azure CRP Certificate Generator | DC=Windows Azure CRP Cert... | 8/10/2018 8:39:41 ... | B918754E14B9236CC3BC2D53... | Personal |
| WMSVC | | | 2/20/2027 9:37:18 ... | 6B0A5A8769DE645E6D1DFFE7... | Personal |

Certificate Information

This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer

*Refer to the certification authority's statement for details.

Issued to: *.sandbox.operations.dynamics.com

Issued by: Microsoft IT SSL SHA2

Valid from 10/29/2016 to 4/29/2018

You have a private key that corresponds to this certificate.

Sitting there in plain sight was a valid TLS certificate for the common name *.sandbox.operations.dynamics.com **and the corresponding private key—by the courtesy of Microsoft IT SSL SHA2 CA! This certificate is shared across all sandbox environments, even those hosted for other Microsoft customers. It's used to encrypt the web traffic between the users of the software and the server. All you need to extract this certificate is access to ANY sandbox environment.**

I still could not believe my eyes, so clearly the next goal was to export the private key to make sure it's actually possible to export it and use it outside the system. In Windows the private keys are marked as non-exportable by default and the Certificate Manager refuses to export those. A short C++ program hooking the internal certificate API functions called to check whether a certificate is exportable and a couple minutes later, I had the private key in my hands.

The implications of this are far reaching: an attacker, which has the ability to listen and/or interject himself between the connection from the user to the server (man in the middle), can impersonate the server and thus read all communication in clear text. Furthermore an attacker can modify the communication and thus insert malicious content. Since the attacker can use the original TLS certificate, there's no warning or error on the client side. Just the green padlock indicating a secure connection. The users of this user acceptance (sandbox) systems are high value targets. They are usually in key positions at the respective organisation and have access to valuable information. The sandbox system itself often also contains sensitive information to make the test more realistic. There is even a feature to copy the production database into the sandbox environment to enable this use case. This opens the door for data theft and industrial espionage.

Since the setup of the sandbox system is a copy of the production setup (with the additional RDP access), it's fair to assume that the same issue exists on the even more critical production system. A quick look at the certificate served by *customername.operations.dynamics.com* reveals, that it's a wild card certificate, too (*.operations.dynamics.com). Since it's possible to deploy code to a production system, it should be possible to deploy a piece of code exfiltrating the wild card certificate. Because there's no way anybody would let me deploy this on a customer live production environment and there's no chance to acquire one of those environments for simple testing (licensing issues, huge upfront investment), my research had to stop there.

With the keys to the kingdom in hand, it was time to contact the Microsoft Security Response Center (MSRC) via PGP-encrypted mail (secure@microsoft.com). While I couldn't believe that the TLS certificate was exposed like this, the communication with Microsoft was even more disturbing:

I've sent this initial successful message on 08/17/17:

> Hello,

> I've noticed a vulnerability in the Microsoft managed Azure hosted Dynamics 365 for Operations environments.

> Each separate customer environment (called AOS in the Dynamics world—Windows Server 2012 R2—IIS, accessible by MS customers via RDP) uses the same wildcard server certificate (including the private key) for the domain *.sandbox.operations.dynamics.com, meaning the service hosted for Acme Inc. at acme.sandbox.operations.dynamics.com uses the same TLS wildcard certificate as Evil Inc. hosted at evil.sandbox.operations.dynamics.com.

> Reproduction steps / possible attack scenario:

> Involved parties:
>
> —Bob working as CFO at ACME Inc, Environment hosted at acme.sandbox.operations.dynamics.com
>
> —Attacker Eve trying to steal ACMEs trade secrets working at the competitor Evil Inc, Environment hosted at evil.sandbox.operations.dynamics.com

> Steps:
>
> 1. Attacker Eve visits lcs.dynamics.com and acceses the Azure VM hosting the environment of Evil Inc using the provided RDP file and Administrator credentials—this is the intended way to access your own environment.
> 2. Attacker Eve uses the tool “mimikatz” on this VM to export the public and private key pair of the wildcard certificate for *.sandbox.operations.dynamics.com—Eve has rightfully full administrative access on the hosted azure environment so this is no issue
> 3. Attacker Eve uses this certificate for a man in the middle attack against Bob and poses as acme.sandbox.dynamics.com
> 4. Bob access the acme.sandbox.operations.dynamics.com environment, the traffic gets MITMd by Eve. The attacker has full access to the data being transmitted. Since Dynamics 365 for Operations is an ERP System, it’s a high value target. Bob doesn’t see any difference, since the certificate in use is a legitimate one.

> Mitigation:
>
> During the creation of the environment issue an individual certificate unique for each customer environment.

> I haven’t checked the production environment because of legal and ethical reasons, however visiting acme.operations.dynamics.com also is being served with a wild card certificate issued for *.operations.dynamics.com. Since Eve could deploy code to the production environment, it should be possible to extract this certificate, too.

> Please let me know if i can be of any assistance. Furthermore, please keep me updated about the status.

> Best regards,
> Matthias Gliwka

Later I’ve sent a follow up email containing an encrypted copy of the previously extracted private key.

Having not received a initial response informing me that somebody at Microsoft received my report, three days later I’ve sent a follow up mail asking for confirmation, that they got my report. Five days after my initial report I receive this answer:

> Hello,

> Thank you for following up on this thread. From how I am interpreting this report, it sounds as though the attacker has already received or bypassed admin credentials. As such, this typically would not meet the bar for security servicing.

> Can you provide a scenario where the attacker does not have, or did not bypass, admin credentials—or, a case where the admin credentials were somehow stolen?

> Regards,

> MSRC

I’ve replied with a more detailed explanation of the problem, but until today never got any response on this email thread. Anticipating that I would never get any answer on this thread (which proved to be true) I tried to reach out to an individual working at the PKI Operations team inside Microsoft, which manages the public CA and compliance work. I’ve sent out my mail detailing the problem on 08/23/17 in hope that PKI Operations could reach out to MSRC and make them aware of this issue. On the same day I've got a very friendly response notifying me that he's reaching out to MSRC. A day later I get this response:

> Wanted to update you that I've been chatting with a senior manager in MSRC. He isn't able to find your case based on the number you gave below. Any chance you have the actual case number?

I did not yet receive a case number from the MSRC team, all I had until now was only the CRM ticket number in the subject line. Did I maybe make a mistake while copy-pasting the CRM ticket number in the previous mail? To make sure it's the correct number I forwarded the reply I've got from the MSRC team (see above) with the ticket number in the subject line to the individual at PKI Operations. He informed me, that the MSRC team could not find my mail. So we agreed that I would send a neI did not yet receive a case number from the MSRC team, all I had until now was only the CRM ticket number in the subject line. Did I maybe make a mistake while copy-pasting the CRM ticket number in the previous mail? To make sure it's the correct number I forwarded the reply I've got from the MSRC team (see above) with the ticket number in the subject line to the individual at PKI Operations. He informed me, that the MSRC team could not find my mail. So we agreed that I would send a new mail to the MSRC mailbox (secure@microsoft.com) from a different mail address.w mail to the MSRC mailbox (secure@microsoft.com) from a different mail address.

This time around I actually got a case number a few hours later:

> Thank you very much for your report.

> I have opened case 40397 and the case manager, Sean will be in touch when there is more information.

> […]

> If at any time you have questions or want to share more information, please respond to this message.

> Regards,

> Microsoft Security Response Center

With the new case number I've contacted the very helpful individual at PKI Operations and two days later got this response:

> Update, the folks at MSRC still aren't able to find this case, but the manager involved a bunch more people, so we are all looking into this 😊

A day later the individual at PKI Operations informed me that MSRC has found the mail and is *actively engaged*.

Since I still haven't heard from MSRC a week later, I've continued to follow up on a weekly basis.

**To be clear: I did not expect resolution of the problem within a couple of weeks, all I wanted was a simple response like "Yep, we've got your mail and a human is looking into it" directly from the MSRC team.**

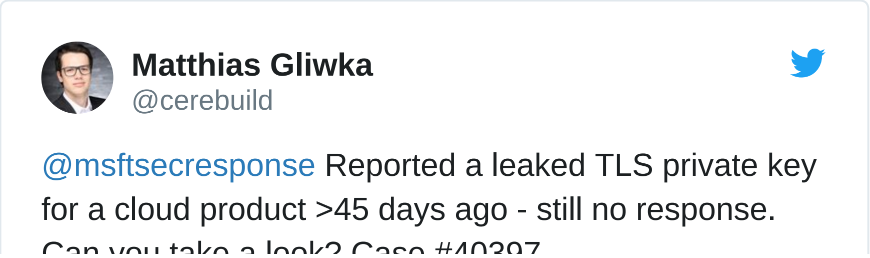
There's so many stories online like this one, where the ticket did not get any attention for years. Since Dynamics 365 for Operations is business critical software and the data transferred over the TLS connection is of very sensitive nature I wanted to make sure that somebody is actually working on this problem.

At the end of September after various follow up mails, I still have not received a single response from MSRC on both threads. So I've sent them an mail detailing that this would be my last attempt to contact them and a full disclosure would trigger, if they would not respond within the next 10 days.

Ten days later having received no response from MSRC out of desperation I've tried to contact the regular Microsoft support using their online chat feature in hope they could get me in touch with someone in the organisation or at least forward a message to them. I've detailed what happened until now and explained, that I'm trying to reach the MSRC team. A few minutes later, I've received this phone number from the support: (562) 981–7600. Could that be the real deal? A call to this number revealed, that it belongs to the Marine Spill Response Corporation (MSRC), ***the largest, dedicated oil spill and***

***emergency response organization in the United States.*** I'm sure a leaked TLS certificate is a serious offense, but is a different kind of leak which needs a different kind of expertise to be handled ;-)

In a last ditch effort I even tried to reach out to them via twitter.



Matthias Gliwka
@cerebuild

@msftsecresponse Reported a leaked TLS private key for a cloud product >45 days ago - still no response. Can you take a look? Case #40397

I've almost given up at this point in time, but to my surprise I received a response on twitter followed by a mail a couple of days later ensuring me that "the servicing team for this endpoint [is] expressly committed to fixing this issue as soon as possible. I will keep you posted with updates."

At this point it should be clear to the reader, that neither of those things happened till this date. The certificate is still out there in use—more than 100 days after the initial discovery. My last follow-up remained unanswered.

The timeline:
08/14/17—Initial discovery of the leaked certificate
08/17/17—First successful contact to MSRC (thread #1)
08/22/17—Response from MSRC, detailing that it doesn't meet the bar for security servicing (thread #1). Last mail from MSRC on this thread.
08/22/17—Mail sent to MSRC detailing why this issue should be dealt with (thread #1)
08/23/17—First mail to individual at PKI Operations
08/23/17—Response from PKI Operations
08/24/17—Response from PKI Operations that MSRC is not able to find my case
08/24/17—Forwarded MSRC mail (thread #1) to PKI Operations
08/25/17—Response from PKI Operations that MSRC is still trying to find the mail
08/25/17—Offered to re-send the mail to MSRC using a different mail

address to PKI Operations
08/25/17—Sent problem description to MSRC again (thread #2)
08/25/17—Received a reply from MSRC with a case number (thread #2). This was the last mail received from MSRC on this thread.
08/26/17—Forwarded mail with the case number to PKI Operations
08/28/17—Received mail from PKI Operations detailing that MSRC was still looking for the new case (thread #2)
08/29/17—Received mail from PKI Operations that MSRC found the mail.
01/09/17—Received mail from individual at PKI Operations that he's dropping off this, because MSRC is "actively engaged"
07/09/17—Follow-up with MSRC (thread #2)—No response
12/09/17—Follow-up with MSRC (thread #2)—No response
18/09/17—Follow-up with MSRC (thread #2)—No response
26/09/17—Follow-up with MSRC (thread #2)—No response
04/10/17—Tweet to @msftsecresponse
10/10/17—Finally got the first response via Twitter + mail
11/15/17—Last follow-up
11/28/17—Full Disclosure