Multiple Vulnerabilities in IGEL Universal Management Suite (UMS) v6.07.100

Vendors

- IGEL

Affected Products

- Universal Management Suite (UMS) v6.07.100

Summary

Superuser/database credentials stored in the HKEY_LOCAL_MACHINE registry

IGEL UMS v6.07.100 stores superuser/database credentials in the `HKEY_LOCAL_MACHINE` registry, which allows a low-privileged attacker with Operating System (OS) access to read the encrypted dbpassword value.

Hardcoded DES key in PrefDBCredentials

A hardcoded DES key in the PrefDBCredentials class in IGEL UMS v6.07.100 allows an attacker with access to an encrypted dbpassword value to decrypt the password and gain superuser/database access to IGEL UMS and its database.

Transmission of Plaintext Lightweight Directory Access Protocol (LDAP) bind credentials

IGEL UMS v6.07.100 exposes Lightweight Directory Access Protocol (LDAP) bind credentials in encrypted and plaintext forms, which allows a remote, authenticated attacker to obtain access to those credentials.

Hardcoded DES key in LDAPDesPWEncrypter

A hardcoded DES key in the `LDAPDesPWEncrypter` class in IGEL UMS v6.07.100 allows an attacker with access to encrypted LDAP bind credentials to decrypt the password and obtain access to plaintext LDAP bind credentials.

Remediation/Mitigation

There are currently no patches to address the findings disclosed here. However, the vendor has provided the following mitigation instructions:

Superuser/database credentials stored in the HKEY_LOCAL_MACHINE registry

Make the UMS Server host accessible only to users that need to access UMS.

Hardcoded DES key in PrefDBCredentials

Keep UMS database and its backups under strict access control.

Transmission of Plaintext Lightweight Directory Access Protocol (LDAP) bind credentials

Only use TLS-secured variant of LDAP access.

Hardcoded DES key in LDAPDesPWEncrypter

Keep UMS database and its backups under strict access control.

Credit

This issue was found by Nick Nam of Atredis Partners

References

- https://nvd.nist.gov/vuln/detail/CVE-2022-25804
- https://nvd.nist.gov/vuln/detail/CVE-2022-25805
- https://nvd.nist.gov/vuln/detail/CVE-2022-25806
- https://nvd.nist.gov/vuln/detail/CVE-2022-25807
- https://github.com/atredispartners/advisories/ATREDIS-2022-0002.md
- https://www.igel.com/igel-solution-family/universal-management-suite/
- https://kb.igel.com/securitysafety/en/isn-2022-13-ums-vulnerabilities-60982590.html

Report Timeline

- 2022-01-05: Atredis Partners sent an initial notification to vendor.
- 2022-01-06: Atredis Partners asks vendor for a status update.
- 2022-02-14: Atredis Partners asks vendor for a status update while notifying of pending CVE submission.
- 2022-02-14: Vendor sends GPG key and requests vulnerability details.
- 2022-02-15: Vendor confirms receipt of initial notification and requests advisory.
- 2022-02-16: Atredis Partners sends a draft advisory to vendor.
- 2022-02-18: Vendor requests a disclosure timeline extension and agrees to extend until May 20, 2022.
- 2022-02-23: Vendor provides update on remediation progress.
- 2022-03-04: Vendor and Atredis Partners discuss possible remediation steps.
- 2022-05-20: Atredis Partners requests remediation and mitigation details for public disclosures.
- 2022-05-20: Vendor confirms that vulnerabilities have not been remediated and provides mitigation instructions.
- 2022-05-23: Atredis published this advisory.

Technical Details

Superuser/database credentials stored in the HKEY_LOCAL_MACHINE registry

The `de.igel.rm.config.PrefDBCredentials` class in `RMGUI.jar` is used to encrypt and decrypt credentials for the UMS superuser. On the Windows operating system, the stored credentials are located in the registry at `HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Prefs\de\igel\rm\config\serverconfig` as `dbuser` and `dbpassword` values.

Hardcoded DES key the `PrefDBCredentials`

`PrefDBCredentials` instantiates a static byte array with 8 bytes and uses that array as the cryptographic key for the encryption and decryption of the `dbpassword` value.

```
    private static byte[] mat = new byte[] { 5, -88, -123, -1, -10, 52, -28, 70 };

    private Encrypter pwCrypt = (Encrypter)new DESEncrypter(mat);
  ...
    public String getPassword() {
      String ePw = ServerConfig.getInstance().getValue(ServerConfigParam.dbuserpasswor
      if (ePw == null)
        return null;
      return this.pwCrypt.decrypt(ePw);
    }
  ...
    public void storePassword(String newPw) {
      try {
        if (newPw == null)
          newPw = "";
        ServerConfig serverConfig = ServerConfig.getInstance();
        if (serverConfig != null) {
          serverConfig.setValue(ServerConfigParam.dbuserpassword, this.pwCrypt.encrypt
          serverConfig.flush();
        }
      } catch (BackingStoreException e) {
        log.error(e.getLocalizedMessage(), e);
      }
    }
```

Transmission of Plaintext Lightweight Directory Access Protocol (LDAP) bind credentials

The `cmd_mgt_load_mgt_tree` command instructs UMS to return data of type `de.igel.rm.mgt.common.data.MgtTreeData` that includes `de.igel.rm.mgt.common.data.ADRootMasterData`. `de.igel.rm.mgt.common.data.ADRootMasterData` contains Active Directory (AD) configuration data including the bind credentials used to search AD.

```
POST /servlet/RMServPut HTTP/1.1
Content-Type: application/json
User-Agent: Java/1.8.0_311
Host: 10.0.0.1:8443
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: close
Content-Length: 115
Cookie: JSESSIONID=8898B3CD46BF8C9686DBFDA246FA8DFD

{
    "host": "10.0.0.1",
    "user": "igelums",
    "password": "the_plaintext_password",
    "command": "cmd_mgt_load_mgt_tree",
    "jackson": ""
}
```

Hardcoded DES key in LDAPDesPWEncrypter

The AD bind password is encrypted using DES and an 8 byte static key that is known to both UMS and the UMS Console.

```
public class LDAPDesPWEncrypter {
  private static final Log log = LogFactory.getLog(LDAPDesPWEncrypter.class);

  private static byte[] keymaterial = new byte[] { 123, -12, -55, 10, -1, 85, -28, -

  public static String encrypt(String pw) {
    if (pw == null)
      return null;
    try {
      return (new DESEncrypter(keymaterial)).encrypt(pw);
    } catch (GeneralSecurityException e) {
      log.error(e);
      return pw;
    }
  }

  public static String decrypt(String storedPw) {
```

```
      if (storedPw == null)
        return null;
      try {
        return (new DESEncrypter(keymaterial)).decrypt(storedPw);
      } catch (GeneralSecurityException e) {
        log.error(e);
        return storedPw;
      }
    }
  }
```

de.igel.rm.ldap.LDAPDesPWEncrypter