Network Vulnerability Assessment

Objective:

Identify and mitigate Network Vulnerabilities.

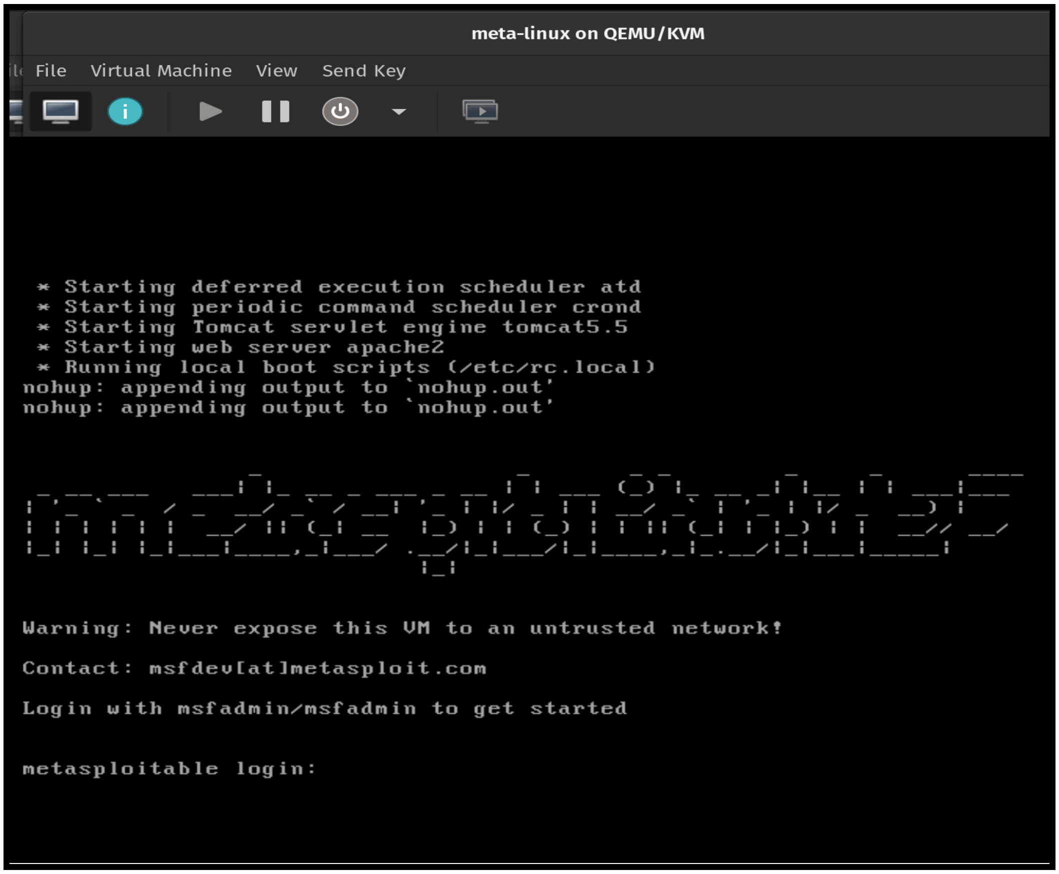
Tools:

OpenVAS software, Kali Linux, Metasploitable-2(Linux) vulnerable VM

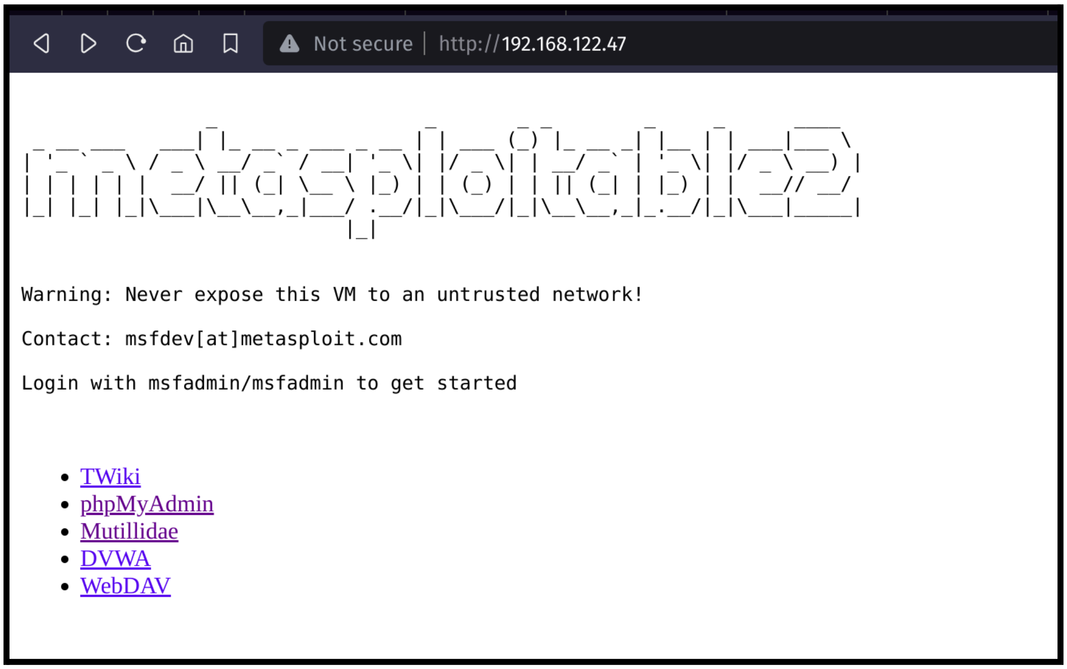
Procedure:

1) Setup the vulnerable VM for network scanning

Download from here: Metasploitable download | SourceForge.net
VM Setup using Virtual Machine Manager: Metasploitable 2 Installing on Kali Linux
`$ ifconfig` on Vulnerable machine



```
 * Starting deferred execution scheduler atd
 * Starting periodic command scheduler crond
 * Starting Tomcat servlet engine tomcat5.5
 * Starting web server apache2
 * Running local boot scripts (/etc/rc.local)
nohup: appending output to `nohup.out'
nohup: appending output to `nohup.out'


                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login:
```



```
                _                  _       _ _        _     _      ____  
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___|___ \ 
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \ __) |
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __// __/ 
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|_____|
                            |_|                                          


Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started
```

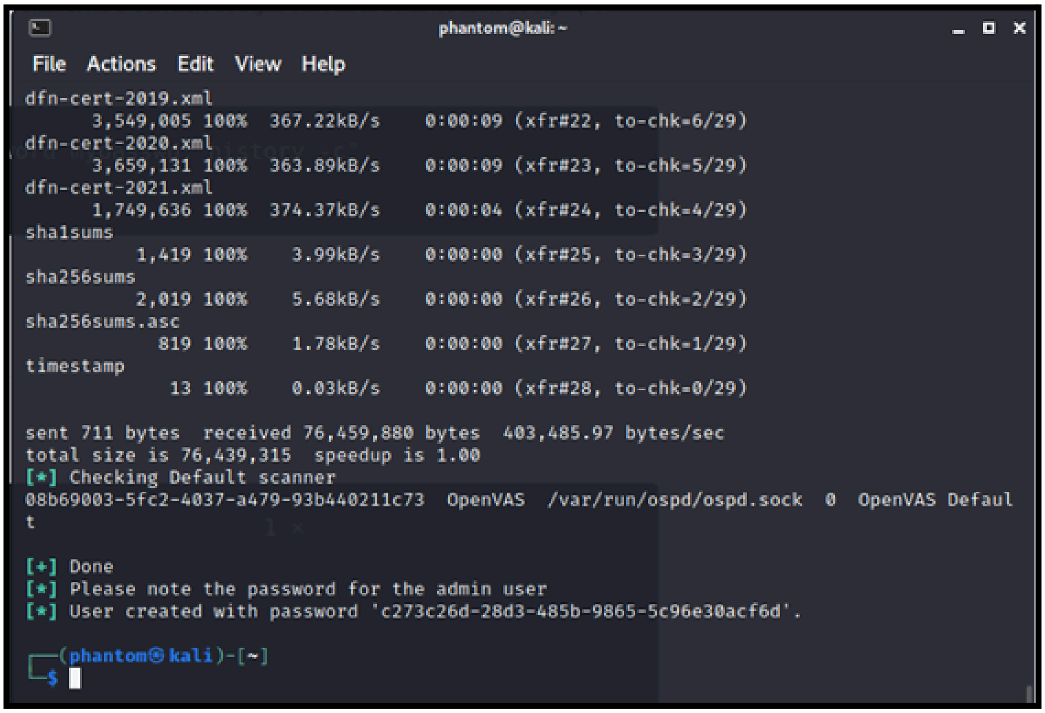
- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

2) Installing OpenVAS

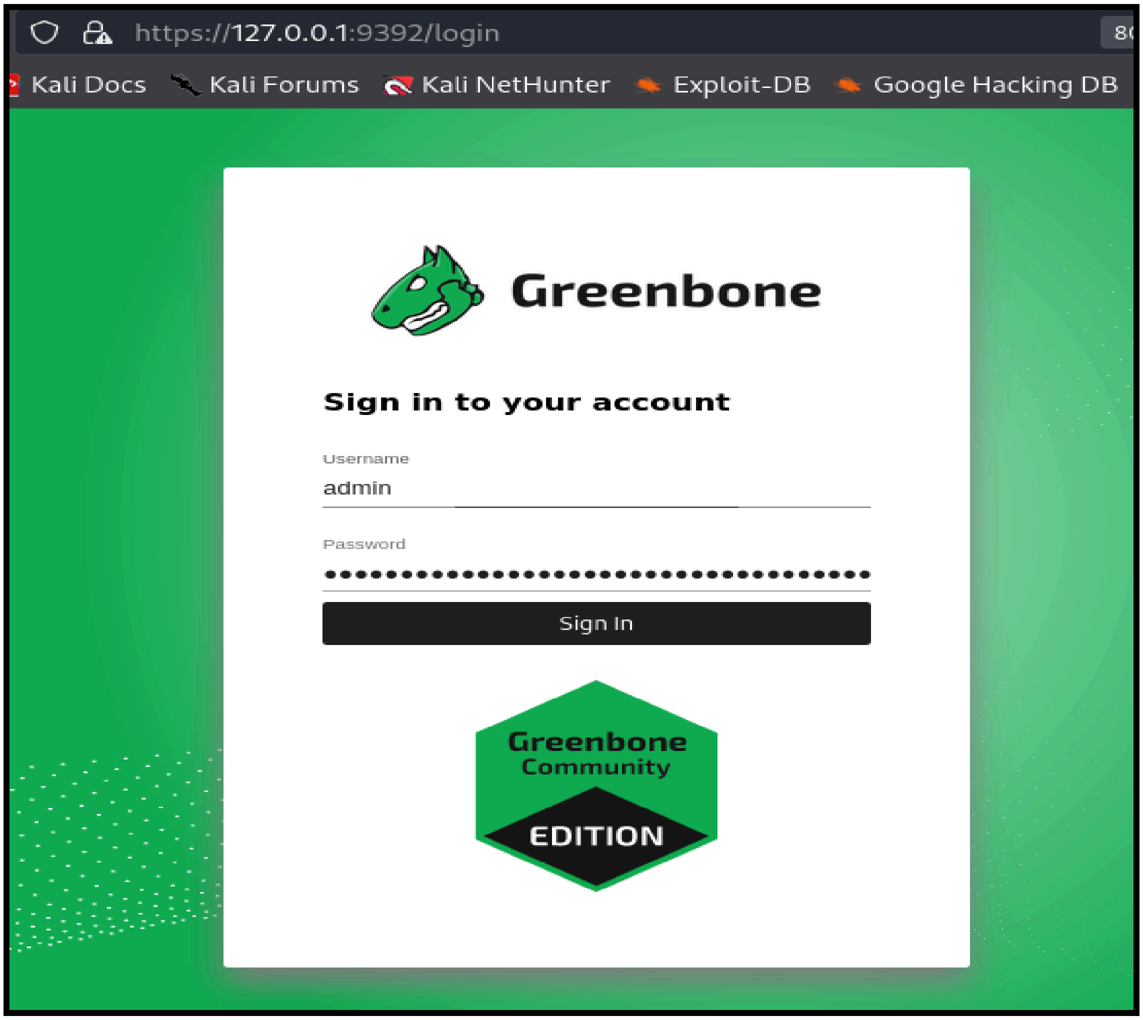
Follow the steps to install: Installing OpenVAS on Kali Linux - GeeksforGeeks

3) Post installation

Open in any browser the url: https://127.0.0.1:9392/login. The username is admin and password can be found after running the cmd: $ gvm-setup in the second-step as seen in the bottom of the picture.



```
dfn-cert-2019.xml
      3,549,005 100%  367.22kB/s    0:00:09 (xfr#22, to-chk=6/29)
dfn-cert-2020.xml
      3,659,131 100%  363.89kB/s    0:00:09 (xfr#23, to-chk=5/29)
dfn-cert-2021.xml
      1,749,636 100%  374.37kB/s    0:00:04 (xfr#24, to-chk=4/29)
sha1sums
          1,419 100%    3.99kB/s    0:00:00 (xfr#25, to-chk=3/29)
sha256sums
          2,019 100%    5.68kB/s    0:00:00 (xfr#26, to-chk=2/29)
sha256sums.asc
            819 100%    1.78kB/s    0:00:00 (xfr#27, to-chk=1/29)
timestamp
             13 100%    0.03kB/s    0:00:00 (xfr#28, to-chk=0/29)

sent 711 bytes  received 76,459,880 bytes  403,485.97 bytes/sec
total size is 76,439,315  speedup is 1.00
[*] Checking Default scanner
08b69003-5fc2-4037-a479-93b440211c73  OpenVAS  /var/run/ospd/ospd.sock  0  OpenVAS Defaul
t

[+] Done
[*] Please note the password for the admin user
[*] User created with password 'c273c26d-28d3-485b-9865-5c96e30acf6d'.

┌──(phantom㉿kali)-[~]
└─$ 
```



https://127.0.0.1:9392/login

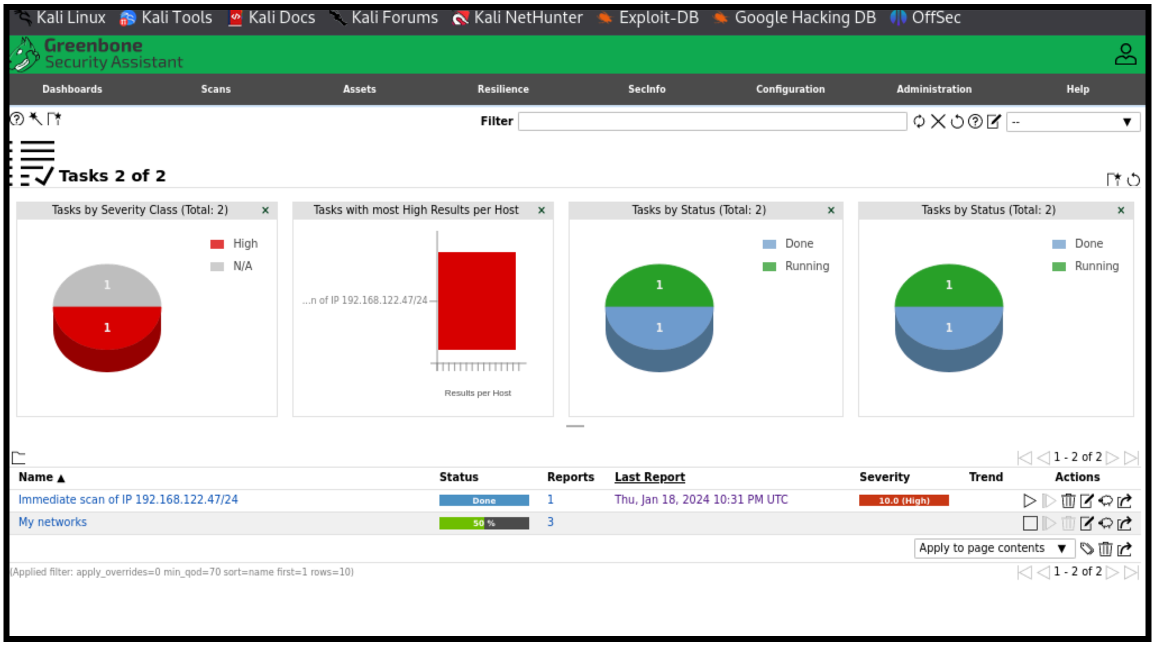
Sign in to your account

Username
admin

Password
●●

4) Scan the IP

Now, we need the ip address of the vulnerable machine to scan the vulnerabilities in the OpenVAS. This step takes some time to complete the scanning. After scanning, reports are generated.



Tasks 2 of 2

1 - 2 of 2

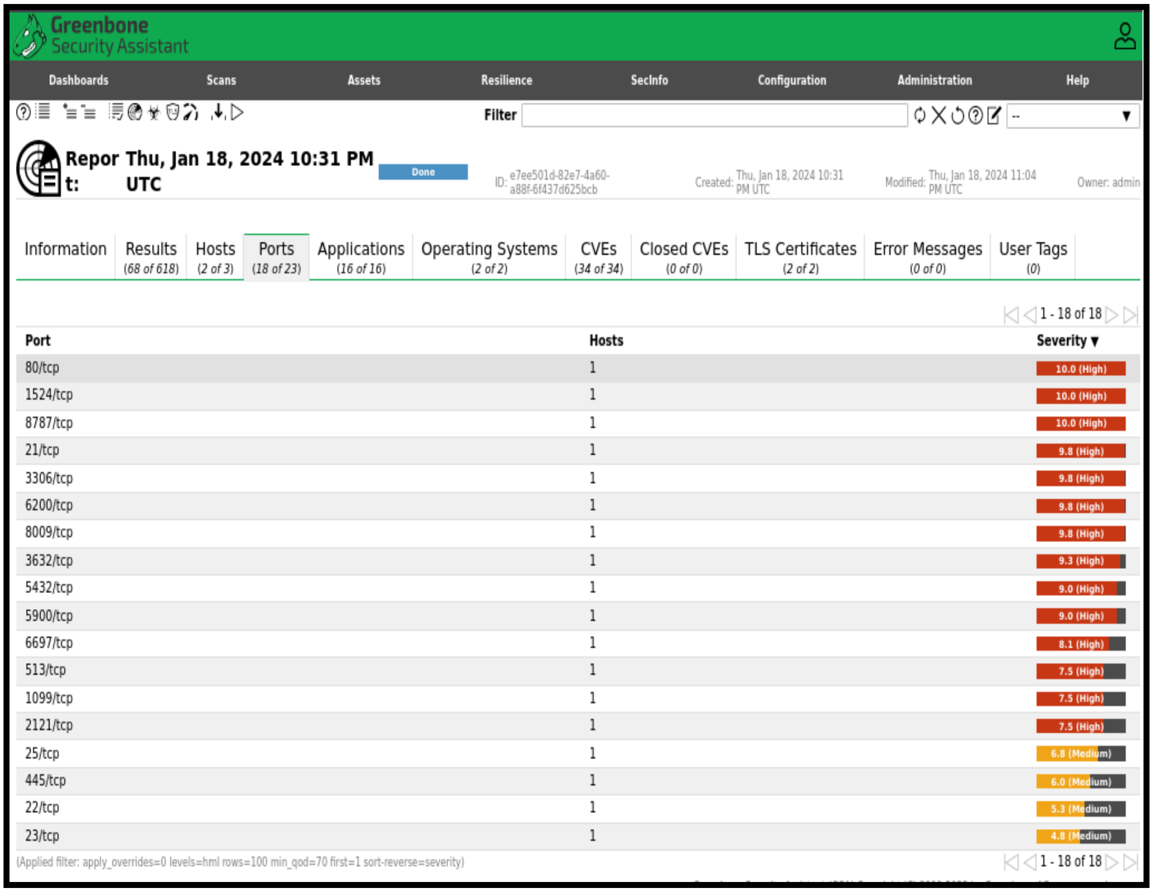
Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
Immediate scan of IP 192.168.122.47/24	Done	1	Thu, Jan 18, 2024 10:31 PM UTC	10.0 (High)		
My networks	50 %	3				

Apply to page contents ▼

(Applied filter: apply_overrides=0 min_qod=70 sort=name first=1 rows=10)

1 - 2 of 2

Some Vulnerable Ports:



Report: Thu, Jan 18, 2024 10:31 PM UTC Done

ID: e7ee501d-82e7-4a60-a88f-6f437d625bcb Created: Thu, Jan 18, 2024 10:31 PM UTC Modified: Thu, Jan 18, 2024 11:04 PM UTC Owner: admin

Information | Results (68 of 618) | Hosts (2 of 3) | Ports (18 of 23) | Applications (16 of 16) | Operating Systems (2 of 2) | CVEs (34 of 34) | Closed CVEs (0 of 0) | TLS Certificates (2 of 2) | Error Messages (0 of 0) | User Tags (0)

1 - 18 of 18

Port	Hosts	Severity ▼
80/tcp	1	10.0 (High)
1524/tcp	1	10.0 (High)
8787/tcp	1	10.0 (High)
21/tcp	1	9.8 (High)
3306/tcp	1	9.8 (High)
6200/tcp	1	9.8 (High)
8009/tcp	1	9.8 (High)
3632/tcp	1	9.3 (High)
5432/tcp	1	9.0 (High)
5900/tcp	1	9.0 (High)
6697/tcp	1	8.1 (High)
513/tcp	1	7.5 (High)
1099/tcp	1	7.5 (High)
2121/tcp	1	7.5 (High)
25/tcp	1	6.8 (Medium)
445/tcp	1	6.0 (Medium)
22/tcp	1	5.3 (Medium)
23/tcp	1	4.8 (Medium)

(Applied filter: apply_overrides=0 levels=hml rows=100 min_qod=70 first=1 sort-reverse=severity)

1 - 18 of 18

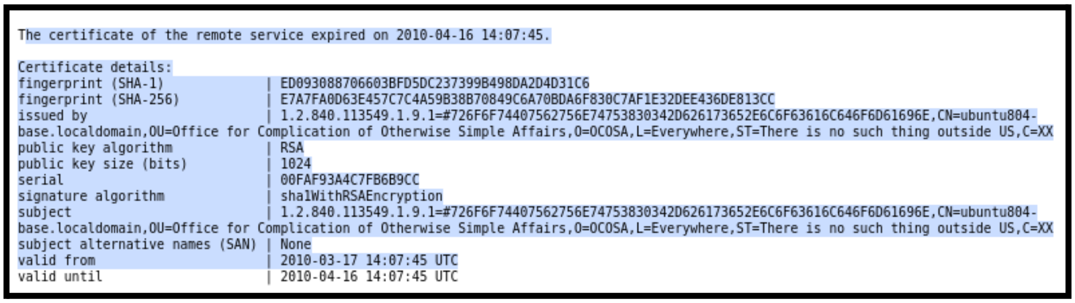
Vulnerabilities:

1) MySQL/Maria default credentials(MySQL Protocol)

- Quality of detection: 95%
- Severity: High - 9.8/10
- Detailed report - Possible to login as root with an empty password. Even brute force attack was possible because of weak credentials
- Mitigation/solution: Change passwords as soon as possible. Contact the vendor for updates, the software is of obsolete version

2) SSL/TLS certificate expired

- Quality of detection: 99%
- Severity: Medium 5/10
- Detailed report: This script checks expiry dates of certificates associated with SSL/TLS-enabled services on the target and reports whether any have already expire



```
The certificate of the remote service expired on 2010-04-16 14:07:45.

Certificate details:
fingerprint (SHA-1)             | ED093088706603BFD5DC237399B498DA2D4D31C6
fingerprint (SHA-256)           | E7A7FA0D63E457C7C4A59B38B70849C6A70BDA6F830C7AF1E32DEE436DE813CC
issued by                       | 1.2.840.113549.1.9.1=#726F6F74407562756E74753830342D626173652E6C6F63616C646F6D61696E,CN=ubuntu804-
base.localdomain,OU=Office for Complication of Otherwise Simple Affairs,O=OCOSA,L=Everywhere,ST=There is no such thing outside US,C=XX
public key algorithm            | RSA
public key size (bits)          | 1024
serial                          | 00FAF93A4C7FB6B9CC
signature algorithm             | sha1WithRSAEncryption
subject                         | 1.2.840.113549.1.9.1=#726F6F74407562756E74753830342D626173652E6C6F63616C646F6D61696E,CN=ubuntu804-
base.localdomain,OU=Office for Complication of Otherwise Simple Affairs,O=OCOSA,L=Everywhere,ST=There is no such thing outside US,C=XX
subject alternative names (SAN) | None
valid from                      | 2010-03-17 14:07:45 UTC
valid until                     | 2010-04-16 14:07:45 UTC
```

- Mitigation/solution:

  Replace the SSL/TLS certificate with the new one

3) FTP Brute force logins reporting

- Quality of detection: 95%
- Severity: Medium 7.5/10
- Detailed report: It was possible to login into the remote FTP server using weak/known credentials.

  Same credentials:

  Msfadmin:msfadmin

  Postgres:postgres

  Service:service

  User:user

- Mitigation/solution: Change passwords, do not enter username and passwords as same

4) SSL/TLS openssl ccs man in the middle security bypass vulnerability

Quality of detection: 80%

Severity: High 8.1/10

Detailed report: UnrealIRCd is prone to authentication spoofing vulnerability. The flaw exists due to an error in the 'm_authenticate' function in 'modules/m_sasl.c' script. Successful exploitation of this vulnerability will allows remote attackers to spoof certificate fingerprints and consequently log in as another user.

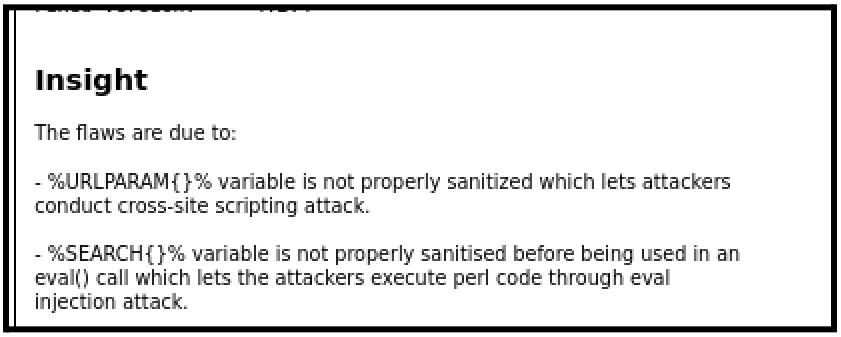
Mitigation/solution: Upgrade to version 4 or above.

5) TWiki XSS and command execution vulnerabilities

Quality of detection: 80%

Severity: High 10/10

Detailed report:TWiki is prone to Cross-Site Scripting (XSS) and Command Execution Vulnerabilities. Successful exploitation could allow execution of arbitrary script code or commands. This could let attackers steal cookie-based authentication credentials or compromise the affected application.



Insight

The flaws are due to:

- %URLPARAM{}% variable is not properly sanitized which lets attackers conduct cross-site scripting attack.

- %SEARCH{}% variable is not properly sanitised before being used in an eval() call which lets the attackers execute perl code through eval injection attack.

Mitigation/solution: Upgrade to version 4 or later