Esercizio Unit2 S5/L3

Obiettivo:

Effettuare un Vulnerability Scanning sulla macchina Metasploitable utilizzando Nessus, concentrandosi sulle porte comuni. Questo esercizio ha lo scopo di fare pratica con lo strumento Nessus, la configurazione delle scansioni, e di familiarizzare con alcune delle vulnerabilità note.

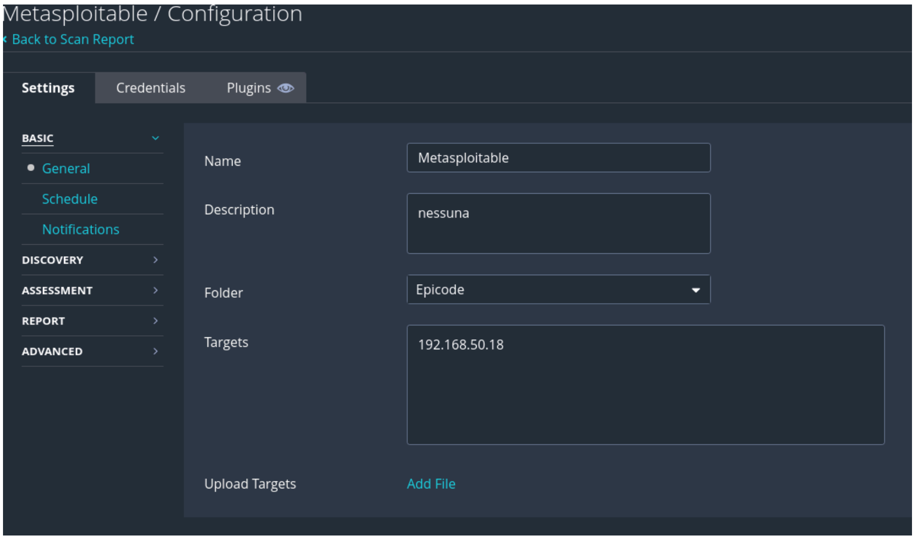
Fasi dell'Esercizio:

1. ***Configurazione della Scansione**:*
   - *○ Target: Metasploitable*
   - *○ Porte: Solo le porte comuni (es. 21, 22, 23, 25, 80, 110, 139, 443, 445, 3389)*
   - *○ Tipo di Scansione: scegliere tra "Basic Network Scan" o "Advanced Scan".*
2. ***Esecuzione della Scansione**:*
   ○ Avvia la scansione configurata su Nessus.
   ○ Attendi il completamento della scansione e assicurati che tutte le porte specificate siano state analizzate.
3. *Analisi del Report:*
   ○ Una volta completata la scansione, scarica e analizza il report generato da Nessus.
   ○ Per ogni vulnerabilità riportata: Leggi attentamente la descrizione fornita nel report. Approfondisci ulteriormente utilizzando i link e le risorse suggerite nel report. Cerca ulteriori informazioni sul Web, se necessario.

IPv4 Metasploitable: 192.168.50.18

Esercizio:

Come prima cosa accedo da terminale e avvio Nessus col comando **sudo service nessusd start.** Poi ricerco nella barra URL **https://kali:8834/,** eseguo il login e mi trovo nella pagina iniziale di Nessus. Clicco sulla sezione "scan", e all'interno della cartella "Epicode" configuro uno scan di vulnerabilità per la macchina Metasploitable, con indirizzo IPv4 192.168.50.18.


Metasploitable / Configuration

Name: Metasploitable

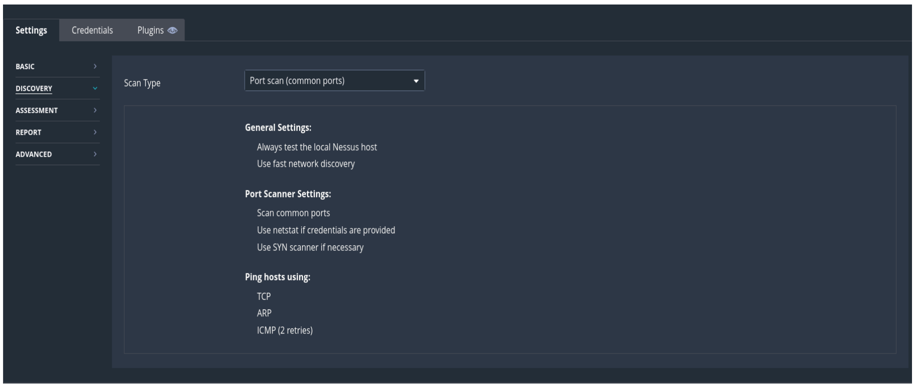
Description: nessuna

Folder: Epicode

Targets: 192.168.50.18

Upload Targets: Add File

decido di non configurare manualmente le porte specifiche per avere uno scan più completo sulla macchina, selezionando nella sezione "discovery" lo scan delle porte comuni, in cui vengono definiti anche i protocolli utilizzati per l'analisi delle vulnerabilità sull'host target.



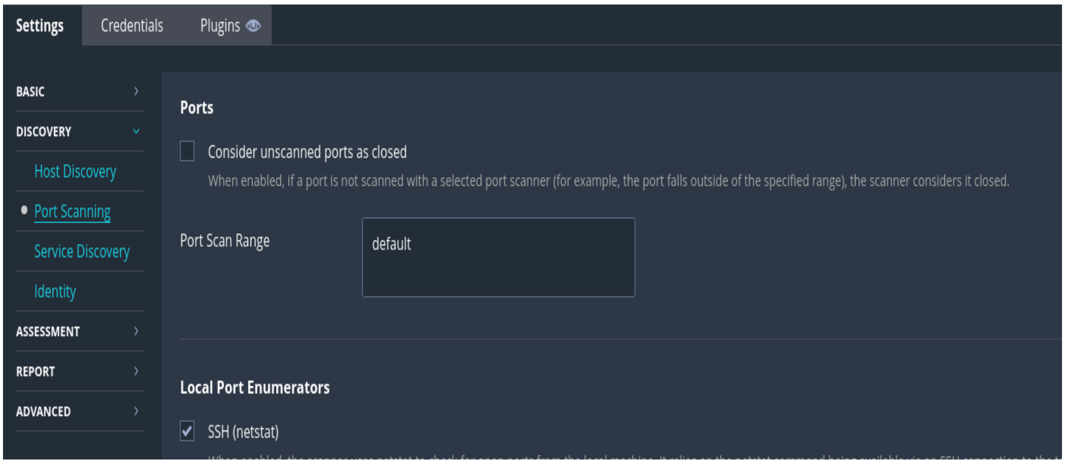
Scan Type: Port scan (common ports)

General Settings:
Always test the local Nessus host
Use fast network discovery

Port Scanner Settings:
Scan common ports
Use netstat if credentials are provided
Use SYN scanner if necessary

Ping hosts using:
TCP
ARP
ICMP (2 retries)

Se avessi voluto selezionare solo determinate porte specifiche avrei potuto farlo sempre in questa sezione, in modalità “custom”, definendole in un range o con una lista separata da virgole nel riquadro sottostante.



Ports

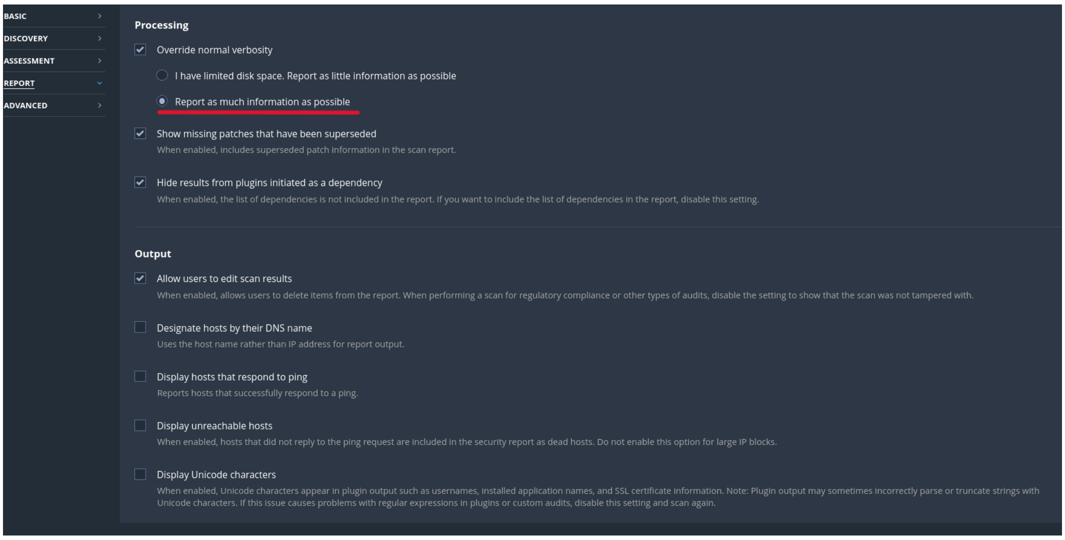
□ Consider unscanned ports as closed
When enabled, if a port is not scanned with a selected port scanner (for example, the port falls outside of the specified range), the scanner considers it closed.

Port Scan Range: default

Local Port Enumerators

☑ SSH (netstat)

Non dando ulteriori specifiche, l’unica altra cosa a cui ho fatto attenzione per avere un report più dettagliato possibile, avendo una sola macchina da analizzare, è stata quella di richiedere un report con più informazioni possibili.



Processing

☑ Override normal verbosity
○ I have limited disk space. Report as little information as possible
● Report as much information as possible

☑ Show missing patches that have been superseded
When enabled, includes superseded patch information in the scan report.

☑ Hide results from plugins initiated as a dependency
When enabled, the list of dependencies is not included in the report. If you want to include the list of dependencies in the report, disable this setting.

Output

☑ Allow users to edit scan results
When enabled, allows users to delete items from the report. When performing a scan for regulatory compliance or other types of audits, disable the setting to show that the scan was not tampered with.

□ Designate hosts by their DNS name
Uses the host name rather than IP address for report output.

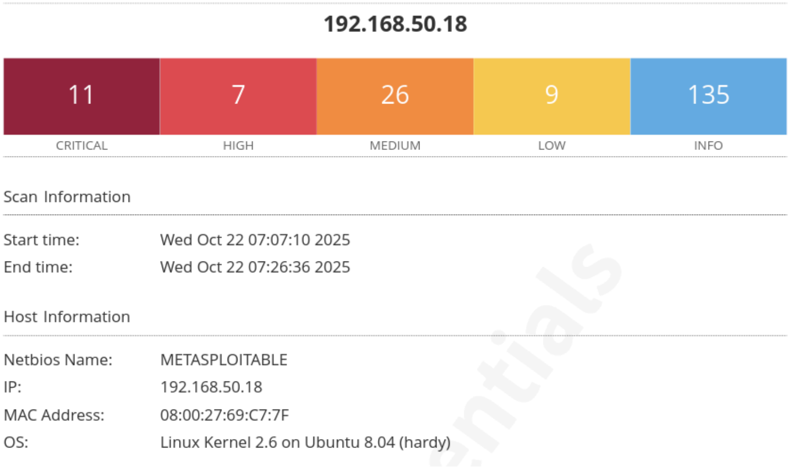
□ Display hosts that respond to ping
Reports hosts that successfully respond to a ping.

□ Display unreachable hosts
When enabled, hosts that did not reply to the ping request are included in the security report as dead hosts. Do not enable this option for large IP blocks.

□ Display Unicode characters
When enabled, Unicode characters appear in plugin output such as usernames, installed application names, and SSL certificate information. Note: Plugin output may sometimes incorrectly parse or truncate strings with Unicode characters. If this issue causes problems with regular expressions in plugins or custom audits, disable this setting and scan again.

Una volta terminata la scansione è stato possibile richiedere un report sia in formato ridotto, che estremamente più dettagliato, in cui viene descritta la tipologia di vulnerabilità

riscontrata. Tutti i report presenteranno all’inizio una rappresentazione grafica del quantitativo di vulnerabilità trovate, divise su più livelli, con diversi colori e numeri in base alla gravità.
Oltre a questo sono presenti informazioni sulla durata dello scan, l’IP della macchina, il sistema operativo presente sulla macchina e il MAC address



192.168.50.18

11	7	26	9	135
CRITICAL	HIGH	MEDIUM	LOW	INFO

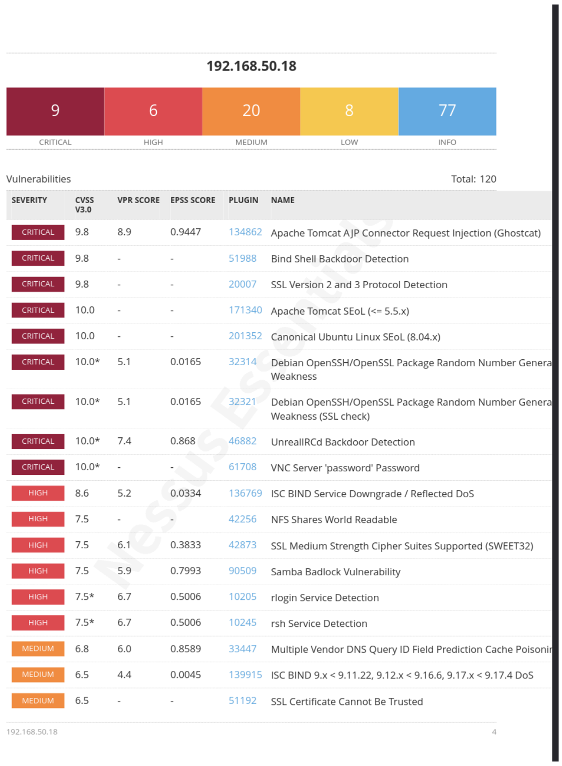
Scan Information

Start time:	Wed Oct 22 07:07:10 2025
End time:	Wed Oct 22 07:26:36 2025

Host Information

Netbios Name:	METASPLOITABLE
IP:	192.168.50.18
MAC Address:	08:00:27:69:C7:7F
OS:	Linux Kernel 2.6 on Ubuntu 8.04 (hardy)

Per mostrare le differenze ho richiesto entrambe le versioni possibili del report. Una più **versione semplificata**, in cui si mostrano in maniera sommaria la lista delle problematiche col proprio nome, il grado di severità sulla scala CVSS, da 1 a 10, e il colore, dal rosso acceso al blu.
Cliccando sul numero presente nella colonna “plugin” verremo portati sul sito Nessus di tenable.com, dove sarà descritta la problematica in maniera più specifica.

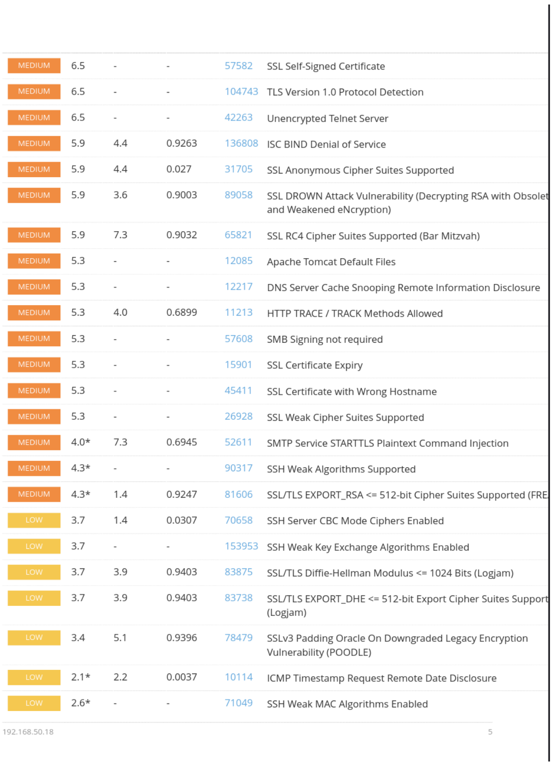


192.168.50.18

9	6	20	8	77
CRITICAL	HIGH	MEDIUM	LOW	INFO

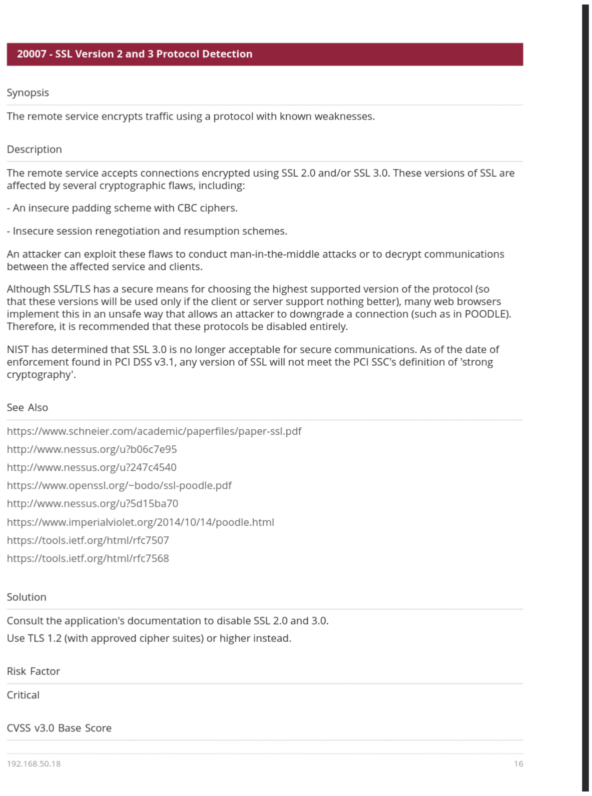
Vulnerabilities Total: 120

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
CRITICAL	9.8	8.9	0.9447	134862	Apache Tomcat AJP Connector Request Injection (Ghostcat)
CRITICAL	9.8	-	-	51988	Bind Shell Backdoor Detection
CRITICAL	9.8	-	-	20007	SSL Version 2 and 3 Protocol Detection
CRITICAL	10.0	-	-	171340	Apache Tomcat SEoL (<= 5.5.x)
CRITICAL	10.0	-	-	201352	Canonical Ubuntu Linux SEoL (8.04.x)
CRITICAL	10.0*	5.1	0.0165	32314	Debian OpenSSH/OpenSSL Package Random Number Genera Weakness
CRITICAL	10.0*	5.1	0.0165	32321	Debian OpenSSH/OpenSSL Package Random Number Genera Weakness (SSL check)
CRITICAL	10.0*	7.4	0.868	46882	UnrealIRCd Backdoor Detection
CRITICAL	10.0*	-	-	61708	VNC Server 'password' Password
HIGH	8.6	5.2	0.0334	136769	ISC BIND Service Downgrade / Reflected DoS
HIGH	7.5	-	-	42256	NFS Shares World Readable
HIGH	7.5	6.1	0.3833	42873	SSL Medium Strength Cipher Suites Supported (SWEET32)
HIGH	7.5	5.9	0.7993	90509	Samba Badlock Vulnerability
HIGH	7.5*	6.7	0.5006	10205	rlogin Service Detection
HIGH	7.5*	6.7	0.5006	10245	rsh Service Detection
MEDIUM	6.8	6.0	0.8589	33447	Multiple Vendor DNS Query ID Field Prediction Cache Poisonir
MEDIUM	6.5	4.4	0.0045	139915	ISC BIND 9.x < 9.11.22, 9.12.x < 9.16.6, 9.17.x < 9.17.4 DoS
MEDIUM	6.5	-	-	51192	SSL Certificate Cannot Be Trusted



MEDIUM	6.5	-	-	57582	SSL Self-Signed Certificate
MEDIUM	6.5	-	-	104743	TLS Version 1.0 Protocol Detection
MEDIUM	6.5	-	-	42263	Unencrypted Telnet Server
MEDIUM	5.9	4.4	0.9263	136808	ISC BIND Denial of Service
MEDIUM	5.9	4.4	0.027	31705	SSL Anonymous Cipher Suites Supported
MEDIUM	5.9	3.6	0.9003	89058	SSL DROWN Attack Vulnerability (Decrypting RSA with Obsolet and Weakened eNcryption)
MEDIUM	5.9	7.3	0.9032	65821	SSL RC4 Cipher Suites Supported (Bar Mitzvah)
MEDIUM	5.3	-	-	12085	Apache Tomcat Default Files
MEDIUM	5.3	-	-	12217	DNS Server Cache Snooping Remote Information Disclosure
MEDIUM	5.3	4.0	0.6899	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.3	-	-	57608	SMB Signing not required
MEDIUM	5.3	-	-	15901	SSL Certificate Expiry
MEDIUM	5.3	-	-	45411	SSL Certificate with Wrong Hostname
MEDIUM	5.3	-	-	26928	SSL Weak Cipher Suites Supported
MEDIUM	4.0*	7.3	0.6945	52611	SMTP Service STARTTLS Plaintext Command Injection
MEDIUM	4.3*	-	-	90317	SSH Weak Algorithms Supported
MEDIUM	4.3*	1.4	0.9247	81606	SSL/TLS EXPORT_RSA <= 512-bit Cipher Suites Supported (FRE
LOW	3.7	1.4	0.0307	70658	SSH Server CBC Mode Ciphers Enabled
LOW	3.7	-	-	153953	SSH Weak Key Exchange Algorithms Enabled
LOW	3.7	3.9	0.9403	83875	SSL/TLS Diffie-Hellman Modulus <= 1024 Bits (Logjam)
LOW	3.7	3.9	0.9403	83738	SSL/TLS EXPORT_DHE <= 512-bit Export Cipher Suites Support (Logjam)
LOW	3.4	5.1	0.9396	78479	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POODLE)
LOW	2.1*	2.2	0.0037	10114	ICMP Timestamp Request Remote Date Disclosure
LOW	2.6*	-	-	71049	SSH Weak MAC Algorithms Enabled

versione estesa:



20007 - SSL Version 2 and 3 Protocol Detection

Synopsis

The remote service encrypts traffic using a protocol with known weaknesses.

Description

The remote service accepts connections encrypted using SSL 2.0 and/or SSL 3.0. These versions of SSL are affected by several cryptographic flaws, including:

- An insecure padding scheme with CBC ciphers.

- Insecure session renegotiation and resumption schemes.

An attacker can exploit these flaws to conduct man-in-the-middle attacks or to decrypt communications between the affected service and clients.

Although SSL/TLS has a secure means for choosing the highest supported version of the protocol (so that these versions will be used only if the client or server support nothing better), many web browsers implement this in an unsafe way that allows an attacker to downgrade a connection (such as in POODLE). Therefore, it is recommended that these protocols be disabled entirely.

NIST has determined that SSL 3.0 is no longer acceptable for secure communications. As of the date of enforcement found in PCI DSS v3.1, any version of SSL will not meet the PCI SSC's definition of 'strong cryptography'.

See Also

https://www.schneier.com/academic/paperfiles/paper-ssl.pdf

http://www.nessus.org/u?b06c7e95

http://www.nessus.org/u?247c4540

https://www.openssl.org/~bodo/ssl-poodle.pdf

http://www.nessus.org/u?5d15ba70

https://www.imperialviolet.org/2014/10/14/poodle.html

https://tools.ietf.org/html/rfc7507

https://tools.ietf.org/html/rfc7568

Solution

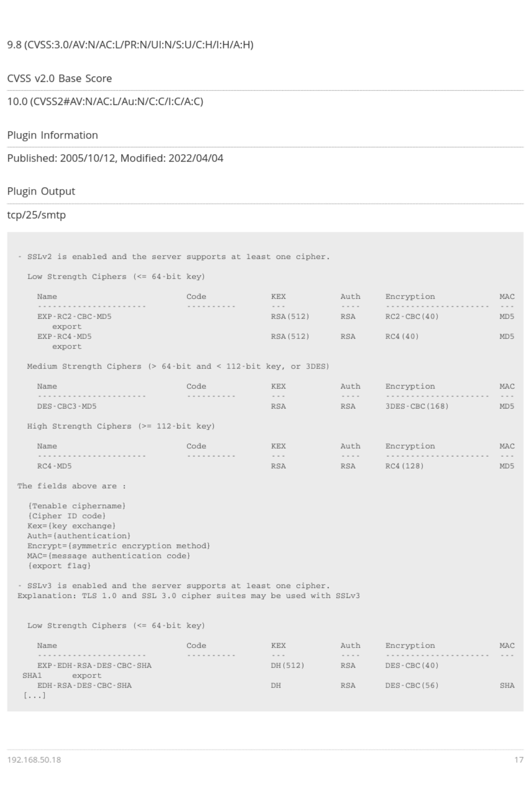
Consult the application's documentation to disable SSL 2.0 and 3.0.

Use TLS 1.2 (with approved cipher suites) or higher instead.

Risk Factor

Critical

CVSS v3.0 Base Score



9.8 (CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

CVSS v2.0 Base Score

10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Plugin Information

Published: 2005/10/12, Modified: 2022/04/04

Plugin Output

tcp/25/smtp

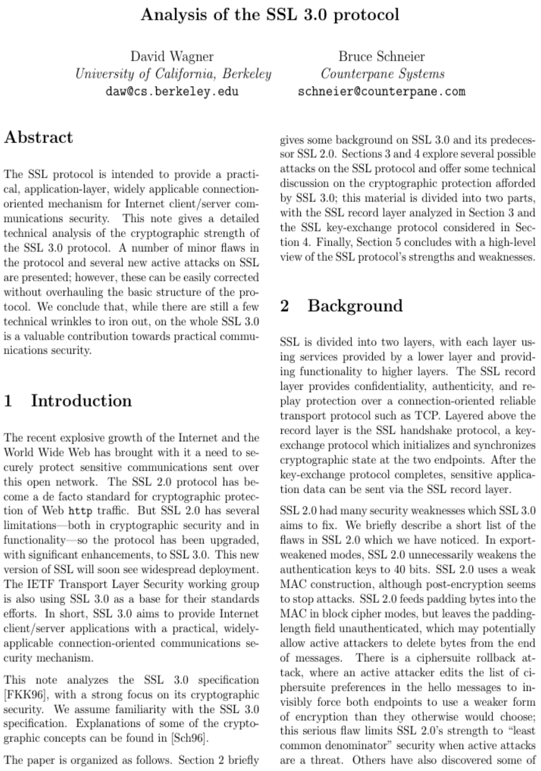
```
- SSLv2 is enabled and the server supports at least one cipher.

  Low Strength Ciphers (<= 64-bit key)

    Name                          Code             KEX           Auth     Encryption             MAC
    ----------------------        ----------       ---           ----     ---------------------  ---
    EXP-RC2-CBC-MD5                                RSA(512)      RSA      RC2-CBC(40)            MD5
       export
    EXP-RC4-MD5                                    RSA(512)      RSA      RC4(40)                MD5
       export

  Medium Strength Ciphers (> 64-bit and < 112-bit key, or 3DES)

    Name                          Code             KEX           Auth     Encryption             MAC
    ----------------------        ----------       ---           ----     ---------------------  ---
    DES-CBC3-MD5                                   RSA           RSA      3DES-CBC(168)          MD5

  High Strength Ciphers (>= 112-bit key)

    Name                          Code             KEX           Auth     Encryption             MAC
    ----------------------        ----------       ---           ----     ---------------------  ---
    RC4-MD5                                        RSA           RSA      RC4(128)               MD5

The fields above are :

  {Tenable ciphername}
  {Cipher ID code}
  Kex={key exchange}
  Auth={authentication}
  Encrypt={symmetric encryption method}
  MAC={message authentication code}
  {export flag}

- SSLv3 is enabled and the server supports at least one cipher.
Explanation: TLS 1.0 and SSL 3.0 cipher suites may be used with SSLv3


  Low Strength Ciphers (<= 64-bit key)

    Name                          Code             KEX           Auth     Encryption             MAC
    ----------------------        ----------       ---           ----     ---------------------  ---
    EXP-EDH-RSA-DES-CBC-SHA                        DH(512)       RSA      DES-CBC(40)
 SHA1      export
    EDH-RSA-DES-CBC-SHA                            DH            RSA      DES-CBC(56)            SHA
 [...]
```

in questo caso si può vedere, prendendo ad esempio una sola delle vulnerabilità presenti nella lista, viene riportata una descrizione dettagliata della problematica, i link con riferimenti esterni per approfondire nello specifico, le soluzioni proposte, lo score sulla scala CVSS, la data in cui è stata scoperta e corretta la vulnerabilità, e nella sezione plugin output si legge il protocollo, la porta e il servizio in utilizzo su questa porta.

Analisi di una vulnerabilità:

Usando come esempio nello specifico la vulnerabilità presentata subito sopra, è possibile quindi farsi un'idea del problema già nella lunga descrizione proposta nel report. Oltre alla descrizione proposta però sono visibili anche vari link che riportano alla documentazione totale del problema.

In questo caso infatti è possibile leggere dal primo link il funzionamento del protocollo SSL 3.0



Analysis of the SSL 3.0 protocol

David Wagner
University of California, Berkeley
daw@cs.berkeley.edu

Bruce Schneier
Counterpane Systems
schneier@counterpane.com

Abstract

The SSL protocol is intended to provide a practical, application-layer, widely applicable connection-oriented mechanism for Internet client/server communications security. This note gives a detailed technical analysis of the cryptographic strength of the SSL 3.0 protocol. A number of minor flaws in the protocol and several new active attacks on SSL are presented; however, these can be easily corrected without overhauling the basic structure of the protocol. We conclude that, while there are still a few technical wrinkles to iron out, on the whole SSL 3.0 is a valuable contribution towards practical communications security.

1 Introduction

The recent explosive growth of the Internet and the World Wide Web has brought with it a need to securely protect sensitive communications sent over this open network. The SSL 2.0 protocol has become a de facto standard for cryptographic protection of Web `http` traffic. But SSL 2.0 has several limitations—both in cryptographic security and in functionality—so the protocol has been upgraded, with significant enhancements, to SSL 3.0. This new version of SSL will soon see widespread deployment. The IETF Transport Layer Security working group is also using SSL 3.0 as a base for their standards efforts. In short, SSL 3.0 aims to provide Internet client/server applications with a practical, widely-applicable connection-oriented communications security mechanism.

This note analyzes the SSL 3.0 specification [FKK96], with a strong focus on its cryptographic security. We assume familiarity with the SSL 3.0 specification. Explanations of some of the cryptographic concepts can be found in [Sch96].

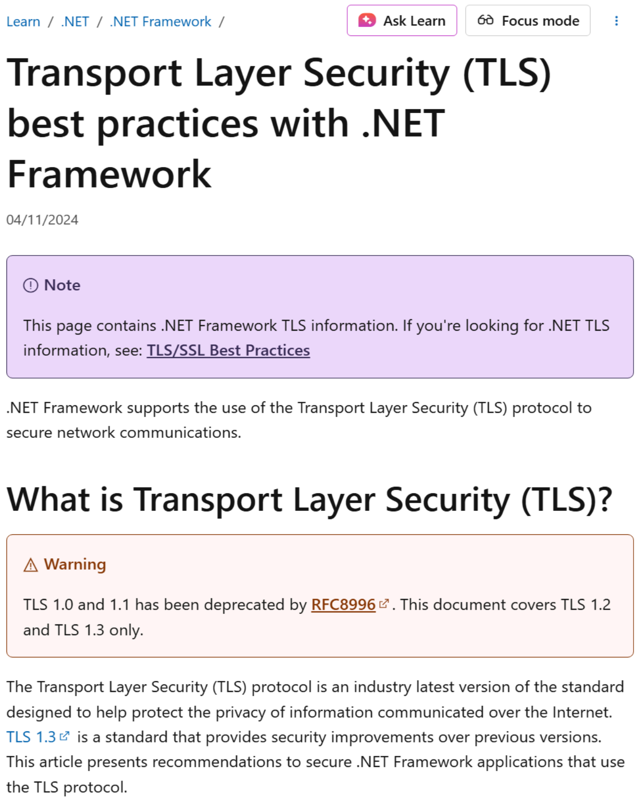
The paper is organized as follows. Section 2 briefly gives some background on SSL 3.0 and its predecessor SSL 2.0. Sections 3 and 4 explore several possible attacks on the SSL protocol and offer some technical discussion on the cryptographic protection afforded by SSL 3.0; this material is divided into two parts, with the SSL record layer analyzed in Section 3 and the SSL key-exchange protocol considered in Section 4. Finally, Section 5 concludes with a high-level view of the SSL protocol's strengths and weaknesses.

2 Background

SSL is divided into two layers, with each layer using services provided by a lower layer and providing functionality to higher layers. The SSL record layer provides confidentiality, authenticity, and replay protection over a connection-oriented reliable transport protocol such as TCP. Layered above the record layer is the SSL handshake protocol, a key-exchange protocol which initializes and synchronizes cryptographic state at the two endpoints. After the key-exchange protocol completes, sensitive application data can be sent via the SSL record layer.

SSL 2.0 had many security weaknesses which SSL 3.0 aims to fix. We briefly describe a short list of the flaws in SSL 2.0 which we have noticed. In export-weakened modes, SSL 2.0 unnecessarily weakens the authentication keys to 40 bits. SSL 2.0 uses a weak MAC construction, although post-encryption seems to stop attacks. SSL 2.0 feeds padding bytes into the MAC in block cipher modes, but leaves the padding-length field unauthenticated, which may potentially allow active attackers to delete bytes from the end of messages. There is a ciphersuite rollback attack, where an active attacker edits the list of ciphersuite preferences in the hello messages to invisibly force both endpoints to use a weaker form of encryption than they otherwise would choose; this serious flaw limits SSL 2.0's strength to "least common denominator" security when active attacks are a threat. Others have also discovered some of

Il secondo link proposto invece va ad analizzare nello specifico invece il funzionamento di TLS e le sue best practice.



Transport Layer Security (TLS) best practices with .NET Framework

04/11/2024

ⓘ Note

This page contains .NET Framework TLS information. If you're looking for .NET TLS information, see: TLS/SSL Best Practices

.NET Framework supports the use of the Transport Layer Security (TLS) protocol to secure network communications.

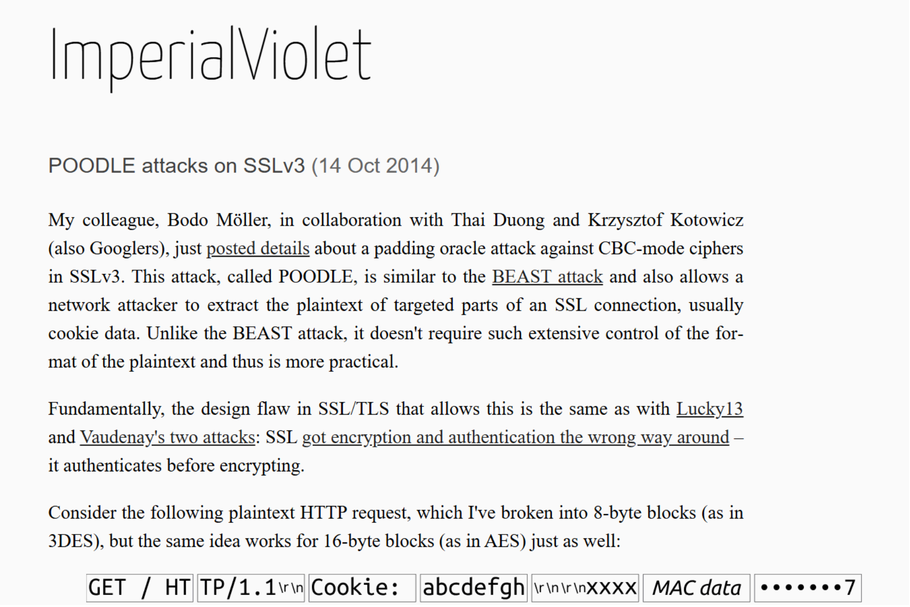
What is Transport Layer Security (TLS)?

⚠ Warning

TLS 1.0 and 1.1 has been deprecated by RFC8996. This document covers TLS 1.2 and TLS 1.3 only.

The Transport Layer Security (TLS) protocol is an industry latest version of the standard designed to help protect the privacy of information communicated over the Internet. TLS 1.3 is a standard that provides security improvements over previous versions. This article presents recommendations to secure .NET Framework applications that use the TLS protocol.

mentre uno degli ultimi link spiega nel dettaglio in che maniera si può sfruttare questa vulnerabilità per eseguire un attacco invece:



ImperialViolet

POODLE attacks on SSLv3 (14 Oct 2014)

My colleague, Bodo Möller, in collaboration with Thai Duong and Krzysztof Kotowicz (also Googlers), just posted details about a padding oracle attack against CBC-mode ciphers in SSLv3. This attack, called POODLE, is similar to the BEAST attack and also allows a network attacker to extract the plaintext of targeted parts of an SSL connection, usually cookie data. Unlike the BEAST attack, it doesn't require such extensive control of the format of the plaintext and thus is more practical.

Fundamentally, the design flaw in SSL/TLS that allows this is the same as with Lucky13 and Vaudenay's two attacks: SSL got encryption and authentication the wrong way around – it authenticates before encrypting.

Consider the following plaintext HTTP request, which I've broken into 8-byte blocks (as in 3DES), but the same idea works for 16-byte blocks (as in AES) just as well:

```
GET / HT|TP/1.1\r\n|Cookie: |abcdefgh|\r\n\r\nxxxx|MAC data|•••••••7
```

Conclusioni:

Nessus è sicuramente uno strumento estremamente utile per questa professione, non solo dando la possibilità di scovare in maniera efficiente le vulnerabilità legate ad un host specifico, ma anche dando informazioni su quest'ultima, fornendo una documentazione dettagliata e proponendo le soluzioni.

Sicuramente bisogna fare attenzione nella configurazione dello scan, soprattutto se si sta cercando di analizzare più host su una rete, perché il processo potrebbe essere estremamente lento e in caso di un computer poco performante si potrebbe rischiare di rovinare l'hardware.