Information security of MONARCH MART

Name of group member:

Name : Nazmul Zaman

ID : 2014755055

Name : Maisha Hossain

ID : 2014755023

Name : Tahmid Bin Zahid

ID: 2014755018

Introduction:

Information and other documents is one of the most prominent assets for Web Server or Website and must be protected from security breach. This paper analyzed the security threats specifically evolve in website network, and with consideration of these issues, proposed information security framework for network environment. The proposed framework reduces the risk of security breach by supporting different phase activities but we will discussed about three important issues ; the first phase assesses the threats and vulnerabilities in order to identify the weak point in website information , the second phase focuses on the highest risk and create actionable remediation plan, the third phase of risk assessment model recognizes the vulnerability management compliance requirement in order to improve local website security position. The proposed framework is applied on Monarch mart (which is the most popular E-commerce website) computing environment and the evaluation result showed the proposed framework enhances the security level of the E-commerce network. This model can be used by risk analyst and security manager of website to perform reliable and repeatable risk analysis in realistic and affordable manner.

Objective:

In this documentation our goal is to find the vulnerability if any website , since our documentation is about an E-commerce website so we chose any popular E-commerce website which is MONAK MART.

We want to make a documentation is about what kind of variability we can face and what kind attacks are we faced by hacker and how can we solve those attacks and our destination is how to build up information security and saving all the database information and client information.

The result of vulnerability of MONARCH MART:

In this case we used some scanning tools which helps us to scan the whole website and gives us the result it means the total situations of this website and what types of vulnerability we found from this website. So the final scanning result it following.:

Vulnerabilities

Scan details

Scan information	
Start url	https://monarchmart.com/
Host	https://monarchmart.com/

Threat level

Acunetix Threat Level 3

One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Alerts distribution

Total alerts found	10
High	1
Medium	3
Low	2
Informational	4

Affected items

Web Server	
Alert group	**TLS 1.0 enabled**
Severity	High
Description	The web server supports encryption through TLS 1.0, which was formally deprecated in March 2021 as a result of inherent security issues. In addition, TLS 1.0 is not considered to be "strong cryptography" as defined and required by the PCI Data Security Standard 3.2(.1) when used to protect sensitive information transferred to or from web sites. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.
Recommendations	It is recommended to disable TLS 1.0 and replace it with TLS 1.2 or higher.
Alert variants	
Details	The SSL server (port: 443) encrypts traffic using TLSv1.0.

Web Server	
Alert group	**TLS 1.1 enabled**
Severity	Medium
Description	The web server supports encryption through TLS 1.1, which was formally deprecated in March 2021 as a result of inherent security issues. When aiming for Payment Card Industry (PCI) Data Security Standard (DSS) compliance, it is recommended to use TLS 1.2 or higher instead. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.
Recommendations	It is recommended to disable TLS 1.1 and replace it with TLS 1.2 or higher.
Alert variants	
Details	The SSL server (port: 443) encrypts traffic using TLSv1.1.

Web Server	
Alert group	**TLS/SSL Sweet32 attack**
Severity	Medium
Description	The Sweet32 attack is a SSL/TLS vulnerability that allows attackers to compromise HTTPS connections using 64-bit block ciphers.
Recommendations	Reconfigure the affected SSL/TLS server to disable support for obsolete 64-bit block ciphers.
Alert variants	
	Cipher suites susceptible to Sweet32 attack (TLS1.0 on port 443):

In this term we found 10 vulnerabilities but we will discuss about high and medium vulnerabilities that we found:

1.High vulnerability which is: TLS 1.0 enabled.

2.Medium vulnerability:

i. TLS 1.1 enabled.

ii. TLS/SSL Sweet32 attack.

TLS:

TLS is a cryptographic protocol that provides end-to-end security of data sent between applications over the Internet. It is mostly familiar to users through its use in secure web browsing, and in particular the padlock icon that appears in web browsers when a secure session is established. However, it can and indeed should also be used for other applications such as e-mail, file transfers, video/audio conferencing, instant messaging and voice-over-IP, as well as Internet services such as DNS and NTP. TLS evolved from Secure Socket Layers (SSL) which was originally developed by Netscape Communications Corporation in 1994 to secure web sessions. SSL 1.0 was never publicly released, whilst SSL 2.0 was quickly replaced by SSL 3.0 on which TLS is based. TLS was first specified in RFC 2246 in 1999 as an applications independent protocol, and whilst was not directly interoperable with SSL 3.0, offered a fallback mode if necessary. However, SSL 3.0 is now considered insecure and was deprecated by RFC 7568 in June 2015, with the recommendation that TLS 1.2 should be used. TLS 1.3 is also currently (as of December 2015) under development and will drop support for less secure algorithms. It should be noted that TLS does not secure data on end systems. It simply ensures the secure delivery of data over the Internet, avoiding possible eavesdropping and/or alteration of the content. TLS is normally implemented on top of TCP in order to encrypt Application Layer protocols such as HTTP, FTP, SMTP and IMAP, although it can also be implemented on UDP, DCCP and SCTP as well (e.g. for VPN and SIP-based application uses). This is known as Datagram Transport Layer Security (DTLS) and is specified in RFCs 6347, 5238 and 6083.

Table.8.Versions of TLS

Major Version	Minor Version	Class
3	1	TLS 1.0
3	2	TLS 2.0
3	3	TLS 3.0

TLS 1.0:

This document presents the latest guidance on rapidly identifying and removing Transport Layer Security (TLS) protocol version 1.0 dependencies in software built on top of Microsoft operating systems, following up with details on product changes and new features delivered by Microsoft to protect your own customers and online services. It is intended to be used as a starting point for building a migration plan to a TLS 1.2+ network environment. While the solutions discussed here may carry over and help with removing TLS 1.0 usage in non-Microsoft operating systems or crypto libraries, they are not a focus of this document.
TLS 1.0 is a security protocol first defined in 1999 for establishing encryption channels over computer networks. Microsoft has supported this protocol since Windows XP/Server 2003. While no longer the default security protocol in use by modern OSes, TLS 1.0 is still supported for backwards compatibility. Evolving regulatory requirements as well as new security vulnerabilities in TLS 1.0 provide corporations with the incentive to disable TLS 1.0 entirely.

Microsoft recommends customers get ahead of this issue by removing TLS 1.0 dependencies in their environments and disabling TLS 1.0 at the operating system level where possible. Given the length of time TLS 1.0 has been supported by the software industry, it is highly recommended that any TLS 1.0 deprecation plan include the following:

1.Code analysis to find/fix hardcoded instances of TLS 1.0 or older security protocols.

2. Network endpoint scanning and traffic analysis to identify operating systems using TLS 1.0 or older protocols.

3. Full regression testing through your entire application stack with TLS 1.0 disabled.

4. Migration of legacy operating systems and development libraries/frameworks to versions capable of negotiating TLS 1.2 by default.

5. Compatibility testing across operating systems used by your business to identify any TLS 1.2 support issues.

6. Coordination with your own business partners and customers to notify them of your move to deprecate TLS 1.0.

7. Understanding which clients may no longer be able to connect to your servers once TLS 1.0 is disabled.

The goal of this document is to provide recommendations which can help remove technical blockers to disabling TLS 1.0 while at the same time increasing visibility into the impact of this change to your own customers. Completing such investigations can help reduce the business

impact of the next security vulnerability in TLS 1.0. For the purposes of this document, references to the deprecation of TLS 1.0 also include TLS 1.1.

Enterprise software developers have a strategic need to adopt more future-safe and agile solutions (otherwise known as Crypto Agility) to deal with future security protocol compromises. While this document proposes agile solutions to the elimination of TLS hardcoding, broader Crypto Agility solutions are beyond the scope of this document.

TLS 1.1:

The web server supports encryption through TLS 1.1, which was formally deprecated in March 2021 as a result of inherent security issues. When aiming for Payment Card Industry (PCI) Data Security Standard (DSS) compliance, it is recommended to use TLS 1.2 or higher instead. According to PCI, "30 June 2018 is the deadline for disabling SSL/early TLS and implementing a more secure encryption protocol – TLS 1.1 or higher (TLS v1.2 is strongly encouraged) in order to meet the PCI Data Security Standard (PCI DSS) for safeguarding payment data.

The Implicit Initialization Vector (IV) is replaced with an explicit IV to protect against Cipher block chaining (CBC) attacks. Handling of padded errors is changed to use the bad_record_mac alert rather than the decryption_failed alert to protect against CBC attacks. IANA registries are defined for protocol parameters Premature closes no longer cause a session to be non-resemble.

TLS 1.2:

The MD5/SHA-1 combination in the pseudorandom function (PRF) was replaced with cipher-suite-specified PRFs. The MD5/SHA-1 combination in the digitally-signed element was replaced with a single hash. Signed elements include a field explicitly specifying the hash algorithm used. There was substantial cleanup to the client's and server's ability to specify which hash and signature algorithms they will accept. Addition of support for authenticated encryption with additional data modes. TLS Extensions definition and AES Cipher Suites were merged in. Tighter checking of EncryptedPreMasterSecret version numbers. Many of the requirements were tightened. Verify data length depends on the cipher suite Description of Bleichenbacher/Dlima attack defenses cleaned up.

Alert message of TLS:

Codes	Alerts	Representations	Types
22	Record_overflow	Payload size exceeded more than $2^{14} + 2048$ bytes	Fatal
48	unknown_ca	CA certificate cannot be trusted or discovered	Fatal
49	accessed_ denied	Negotiation failed due to access control provided by receiver	Fatal
50	decode_ error	Information could not be decoded properly due to incorrect message length	Fatal
51	decrypt_ error	Unable to decrypt the secret key, verify digital signature or authenticity of finished message	Warnin g/ Fatal
60	export_ restriction	Negotiation against export restriction are detected and terminated	Fatal
70	protocol_ version	Protocol version is not supported by server	Fatal
71	insufficient_ security	Handshaking fail due to stronger cipher suite required by server	Fatal
80	internal_error	Error associated to local system and not related to SSL.	Fatal
90	User_cancelled	Abnormal termination of session by user	Fatal
100	no_renegotiation	Client or server response w.r.t hello request is not suitable for renegotiation	Warnin g

Types of Attack:

One of the biggest threats to transport level security due to flaws in SSL/ TLS, which is used to secure the communication between sender and receiver. Vulnerabilities in SSL/TLS triggers both active and passive attacks such as BEAST, CRIME, TIME, BREACH, LUCKY 13, RC4 BIASES, SSL Renegotiation, POODLE, Truncation, Bar Mitzvah etc. and their fixes are listed in table 10.

BEAST attack :

It is the short form of Browser Exploit Against SSL/ TLS attack that occurs by exploiting TLS 1.0 and was developed by T. Duong and J. Riazo. It takes the advantages of symmetric encryption and cipher block chaining (CBC) technique to guess the secret key which is used to encrypt the

plaintext. In TLS 1.0, last ciphertext block is the initialization vector for the current plaintext. XOR operation between initialization vector and plaintext is encrypted by symmetric key to produce corresponding cipher text. If the hacker can guess a plaintext block, he can guess the symmetric key and check whether cipher text is matched or not [4, 5]. It is one type of brute force attack fixed by the corresponding TLS 1.1 and TLS 1.2.

CRIME attack:

It is the short form of Compression Ratio Info Leak Mass Exploitation attack occurs by hijacking the session by decrypting the session cookies in TLS 1.0 and was developed by J. Riazo and T. Duong [6]. It takes the advantages of TLS and SPDY header compression. SPDY is an open networking protocol and controls HTTP traffic developed by Google. Both TLS and SPDY compression techniques use the DEFLATE algorithm, which eliminates duplicate string by compression then encrypts it. The key is obtained by cheating the browser and sending encrypted compressed requests to genuine websites, waiting for the HTTP response size and increasing attack with respect to HTTP responses [7]. Hacker repeats the techniques with different values until the key will be obtained. It is one type of brute force attack fixed by disabling the compression mechanism in TLS 1.1 and TLS 1.2.

Time attack

Timing Info-Leak Made Easy (TIME) attack by which an attacker extracts secret information without eavesdropping into the network and was developed by T. Be"ery and A. Shulman of Imperva. To perform this attack, the hacker wants to know the cookie's location, prefix/suffix and location to insert plaintext. Information about the session cookies is obtained by time taken to get the response from server/ receiver [8]. Due to noise over the network, a single process will be repeated for certain integral number of time and minimal response time is taken as the final response time for that particular requestSuppose client inputs contain “secret element = unknown data” which is the payload and secret element and its value is reflected in the response. In the first iteration for arbitrary user input the response size is 1028 bytes. If in the second iteration the user input is “secret element = a” and the response size is 1008 bytes. So it takes less time compared to the first iteration. With several requests the shortest response time for every character for each position in the payload is computed which happens to be the correct guess and specific value of the secret element.

BREACH attack

Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext is the crime attack against the response body and it was developed by A. Prado, N. Harris and Y. Gluck [9]. Attacker exploits the HTTP compression technique (LZ77 algorithm) by guessing character and symbol without downgrading or tampering SSL to launch this attack and its guess will be reflected in response body [8]. It has taken less than 30 seconds for fairly stable pages to obtain the secret like CSRF token, view state etc... It is vulnerable to any version of SSL or TLS. To launch a breach attack, both attacker and victim must be in the same network. The command and control center has a web server driver called iframe streamer which is going to inject HTTP requests into the victim, callback listener whose work is to call back when a response comes to the victim and traffic monitor observes the length of the cipher text coming back. Basic oracle logic is the collection of algorithms used to guess the secrets. For fighting against Huffman coding, character set pool plus random padding is used and for fighting against block cipher, window technique is used. It is one of the most vulnerable attacks on SSL which is yet to be patched.

LUCKY 13 attack

It is one of the most vulnerable attacks in SSL till now and was developed by N. A. Fardan and K. Paterson at Royal Holloway, University of London in February 2013. It uses padding oracle technique as a side channel attack which is affected on padding of a cipher text. Attackers exploit TLS‟s cipher block chaining by replacing the last some bytes with chosen bytes and watch the amount of time taken by the server to respond [10]. TLS packets that contain true padding take less time to process. If TLS generates a transaction to fail, it produces a message that carries errors which helps the attacker to send malicious packets in a new session repeatedly backing every foregoing failure [6, 11]. Result shows that 223 sessions required extracting information about cookies and 219 sessions required if a 64 bit encoding scheme is used by TLS. Overall LUCKY 13 attack requires 213 sessions;

TLS Truncation Attack

Abnormal termination of TLS connection performed by adversary to keep alive victim session using multiple browser connection [20]. It was developed by B. Smyth and A. Pironti in July 2013. To increase performance, web browser load content through multiple connection. As TLS provides integrity and confidentiality over a single connection, client browser’s multiple connections to

single server are ordered over TLS single connection. Prior to perform this attack, attacker has full control over network which help to inject/ drop packets into different connections. It is triggered at the time of client logout request by injecting TCP FIN or RESET message for that connection prior to it causes request message unavailable to server due to abnormal termination of connection. As logout confirmation comes before the logout request received by the server, attackers launch this attack to keep the session alive without victim knowledge. At last, other connection of browser is used to access victim account and modify it.

Why should we need to enable TLS 1.2 and disable TLS 1.0 and TLS 1.1 :

There are a few reasons why you should disable TLS 1.0 and TLS 1.1 on Windows Server:

1. TLS 1.0 and TLS 1.1 are no longer considered secure, due to the fact that they are vulnerable to various attacks, such as the POODLE attack.
2. Disabling TLS 1.0 and TLS 1.1 on your server will force clients to use a more secure protocol (TLS 1.2), which is less vulnerable to attack.
3. Some government agencies, such as the US National Security Agency (NSA), have recommended that TLS 1.0 and TLS 1.1 be disabled.
4. Microsoft will no longer provide security updates for Windows Server running TLS 1.0 and TLS 1.1.
5. Many major software vendors are phasing out support for TLS 1.0 and TLS 1.1. This includes Google, Microsoft, Mozilla, and Apple.

What is the alternative to TLS 1.0 and TLS 1.1?

The current version of the TLS protocol is TLS 1.3. TLS 1.3 was first defined in 2018, and it includes a number of security improvements over previous versions of the TLS protocol. We suggest you enable TLS 1.2 and TLS 1.3 on your Windows Server instead of TLS 1.0 and TLS 1.1.TLS 1.2 improves upon TLS 1.1 by adding support for Elliptic Curve Cryptography (ECC) and introducing new cryptographic suites that offer better security than the suites used in TLS 1.1. TLS 1.3 improves upon TLS 1.2 by simplifying the handshake process and making it more resistant to man-in-the-middle attacks. In addition, TLS 1.3 introduces new cryptographic suites that offer better security than the suites used in TLS 1.2.

TLS 1.2 and TLS 1.3 are both backward compatible with TLS 1.1 and earlier versions of the protocol. This means that a client that supports TLS 1.2 can communicate with a server that supports TLS 1.1 and vice versa. However, TLS 1.2 and TLS 1.3 are not compatible with each other. A client that supports TLS 1.2 cannot communicate with a server that supports TLS 1.3, and vice versa.

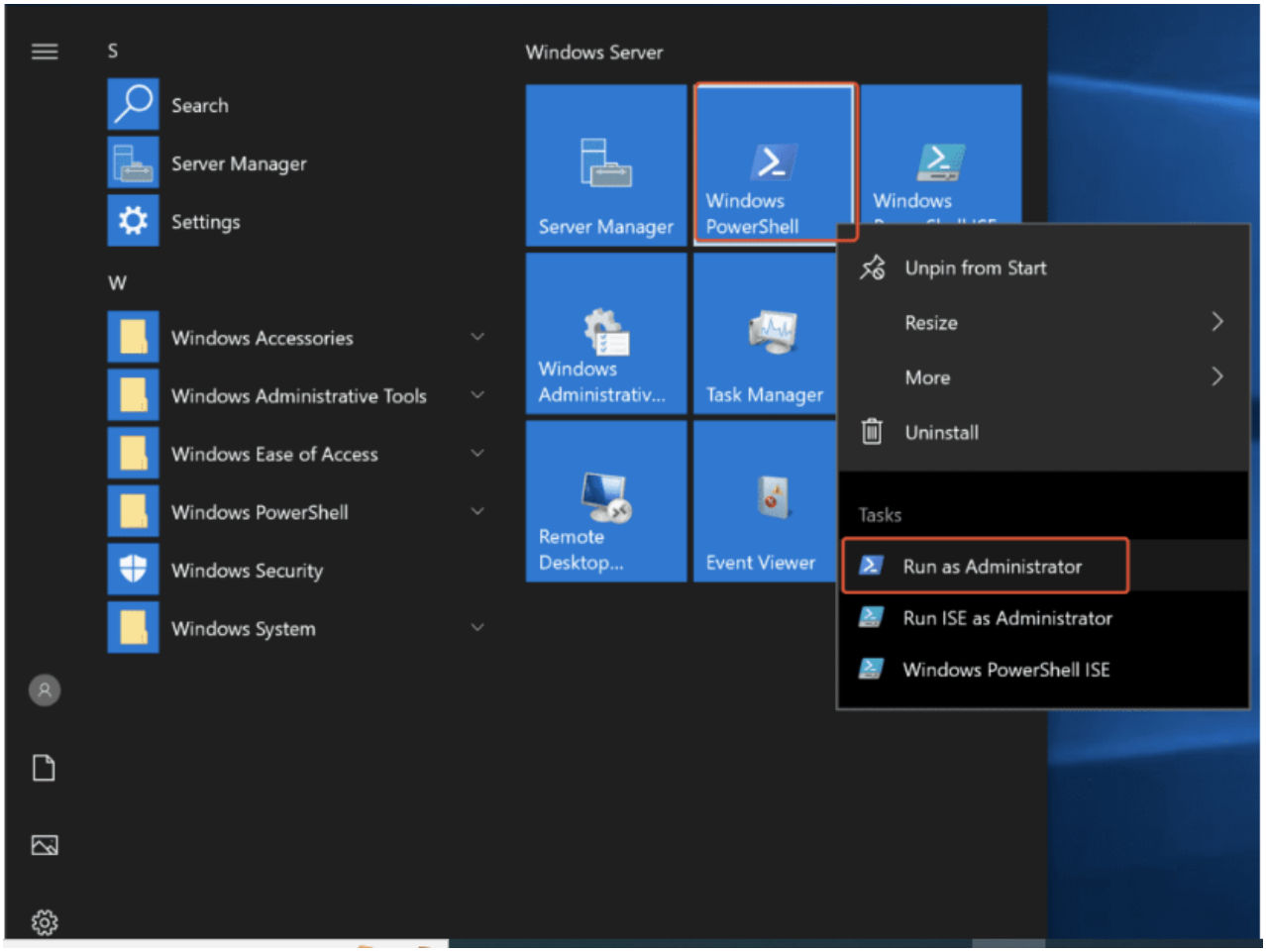
TLS 1.2 is the most widely used version of the TLS protocol, but TLS 1.3 is gaining in popularity. Many major web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge, now support TLS 1.3. In addition, major Internet service providers, such as Cloudflare and Akamai, have started to support TLS 1.3 on their servers. Please visit this page if you want to deeply review the comparison of TLS implementations across different supported servers and clients.

How to Disable TLS 1.0 and TLS 1.1 on Windows Server?

Method 1 : Disable TLS 1.0 and TLS 1.1 using Powershell commands :

Follow this simple procedure to enable TLS 1.2 and TLS 1.2 using Powershell commands.

1.Open Powershell as Administrator

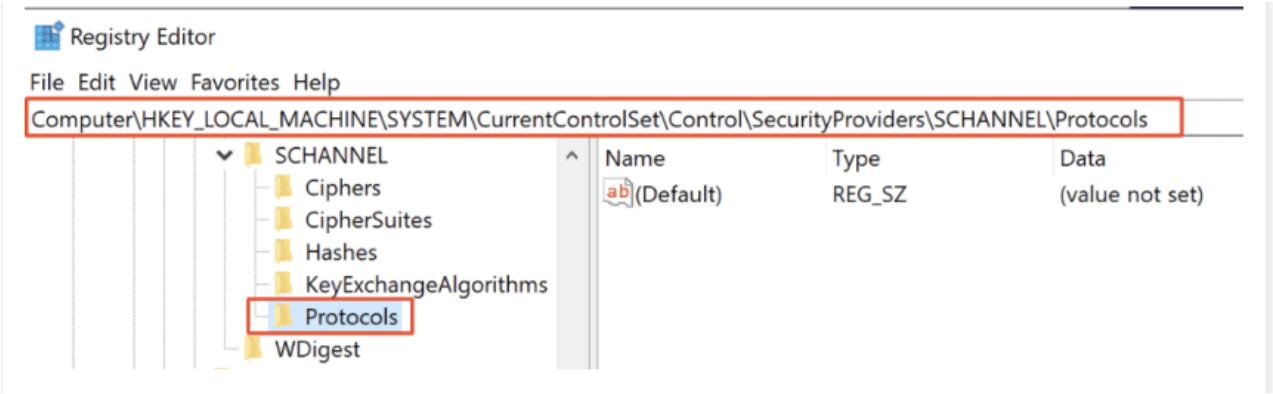


2. Run the below commands to create Registry entries:

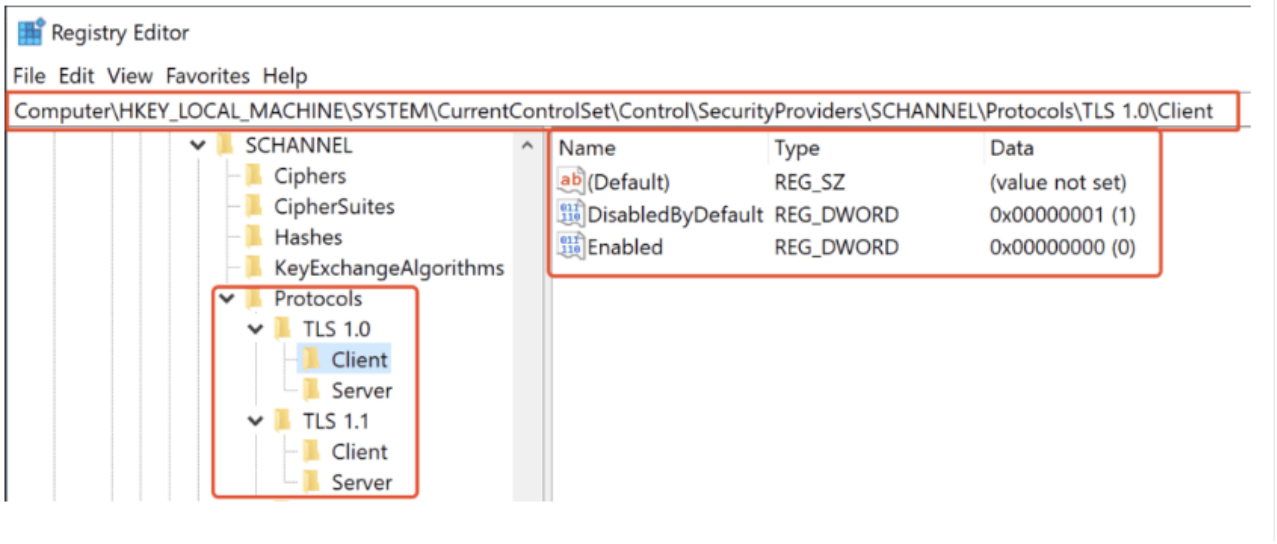
```
- New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.0\Server' -Force
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.0\Server' -PropertyType 'DWORD' -Name 'Enabled' -Value '0'
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.0\Server' -PropertyType 'DWORD' -Name 'DisabledByDefault' -Value '1'

- New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.0\Client' -Force
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.0\Client' -PropertyType 'DWORD' -Name 'Enabled' -Value '0'
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.0\Client' -PropertyType 'DWORD' -Name 'DisabledByDefault' -Value '1'
```

```
- New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.1\Server' -Force
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.1\Server' -PropertyType 'DWORD' -Name 'Enabled' -Value '0'
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.1\Server' -PropertyType 'DWORD' -Name 'DisabledByDefault' -Value '1'

- New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.1\Client' -Force
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.1\Client' -PropertyType 'DWORD' -Name 'Enabled' -Value '0'
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.1\Client' -PropertyType 'DWORD' -Name 'DisabledByDefault' -Value '1'
```

Before running the commands, you can see no items exist underneath the Protocol.



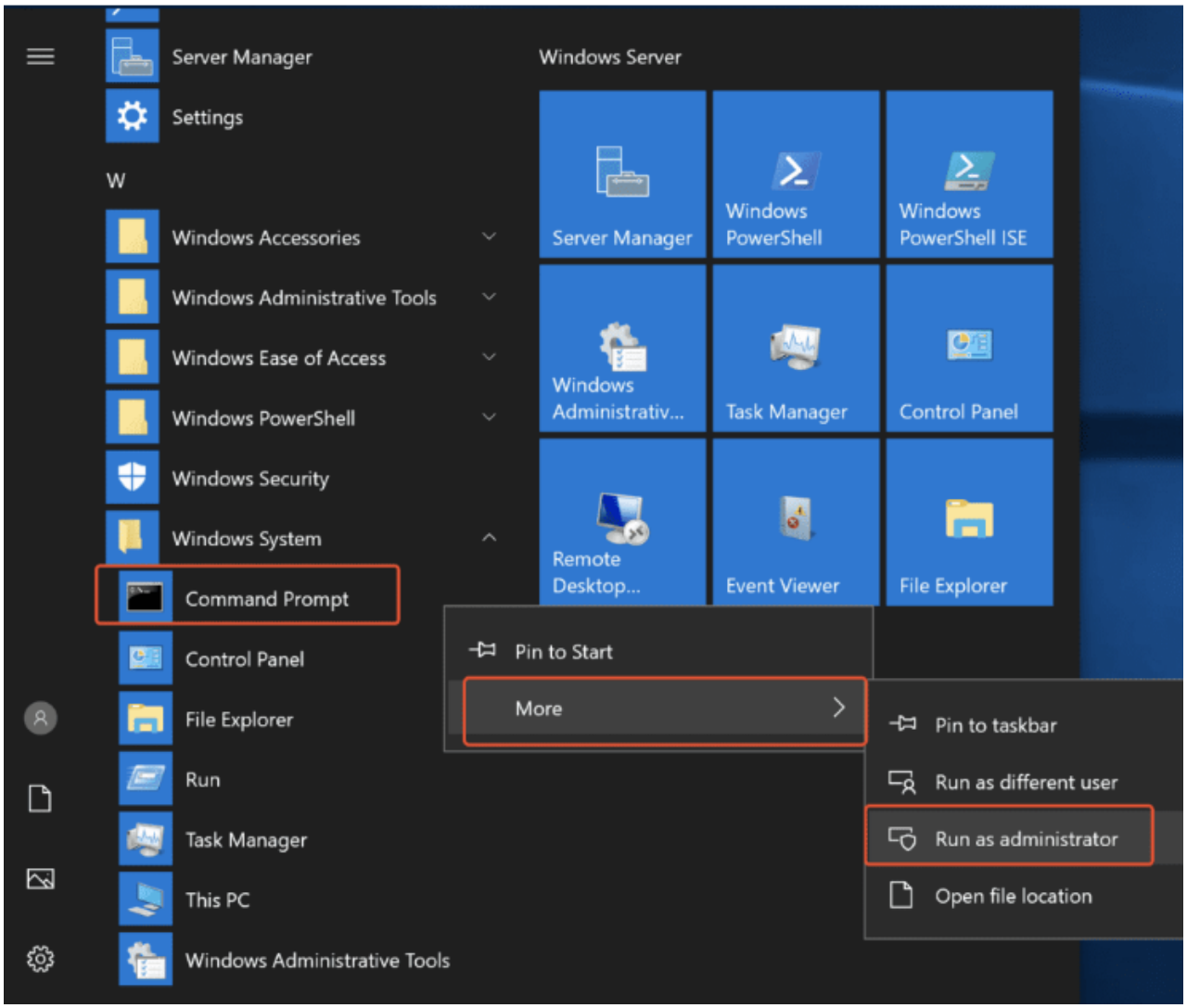
After running the commands you can see there are two keys created ‘TLS 1.0’ & ‘TLS 1.1’, Underneath each protocol there are ‘Client’ &’Server’ Keys inside them there are two items ‘DisableByDefault’ & ‘Enabled’.



Method 2 : Disable TLS 1.0 and TLS 1.1 on Windows Server using CMD:

Follow this simple procedure to disable TLS 1.0 and TLS 1.1 using CMD comments.

1. Open ‘Command Prompt’ as Administrator



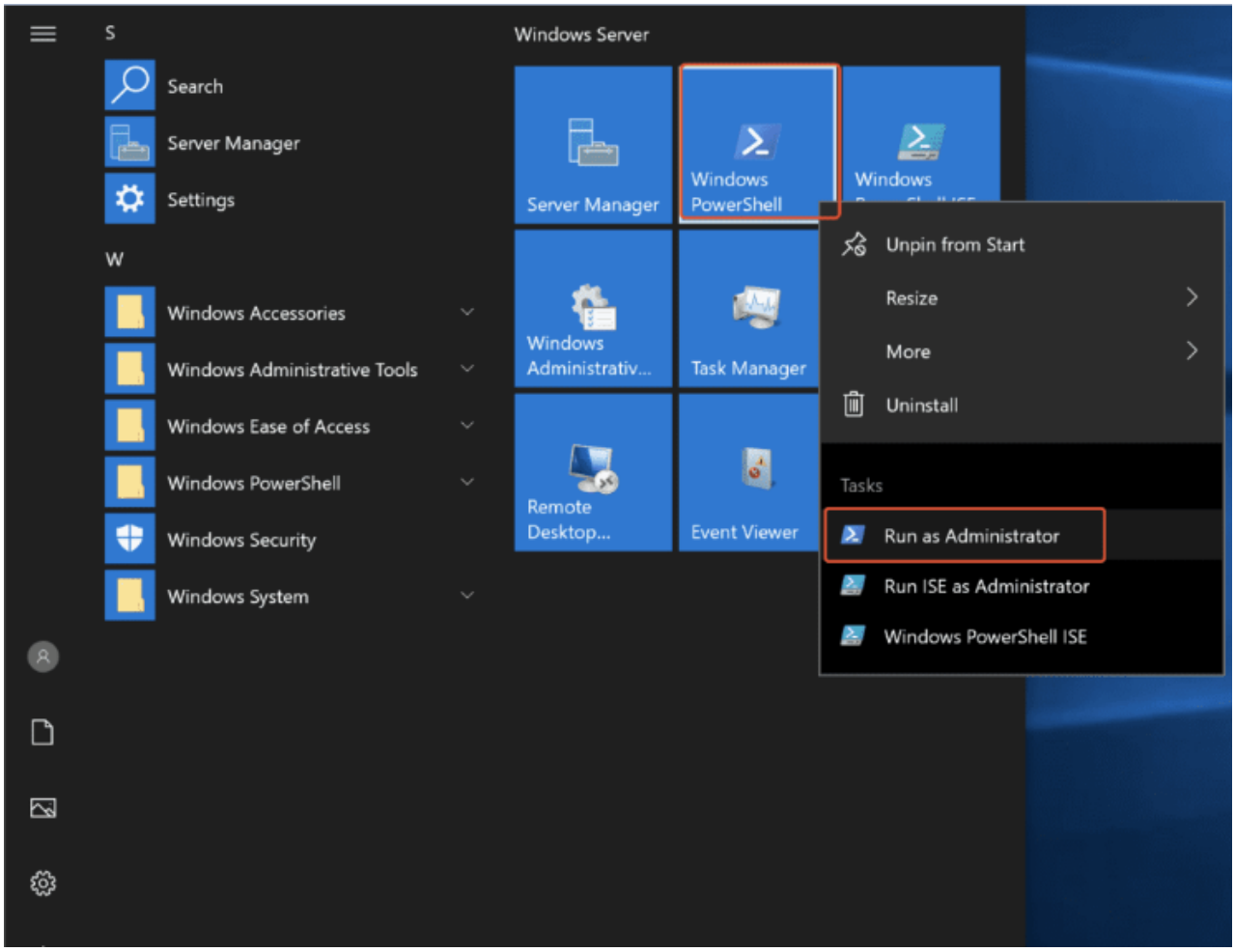
2. Run the below commands to create Registry entries.

```
reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.0\Server" /v Enabled /t REG_DWORD /d 0 /f
reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.0\Server" /v DisabledByDefault /t REG_DWORD /d 1 /f

reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.0\Client" /v Enabled /t REG_DWORD /d 0 /f reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.0\Client" /v DisabledByDefault /t REG_DWORD /d 1 /f



reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.1\Server" /v Enabled /t REG_DWORD /d 0 /f reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.1\Server" /v DisabledByDefault /t REG_DWORD /d 1 /f

reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.1\Client" /v Enabled /t REG_DWORD /d 0 /f reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.1\Client" /v DisabledByDefault /t REG_DWORD /d 1 /f
```

Method 1 : Enable TLS 1.2 and TLS 1.3 on Windows Server using Powershell Commands :

Follow this simple procedure to enable TLS 1.2 and TLS 1.2 using Powershell comments.

1. Open Powershell as Administrator

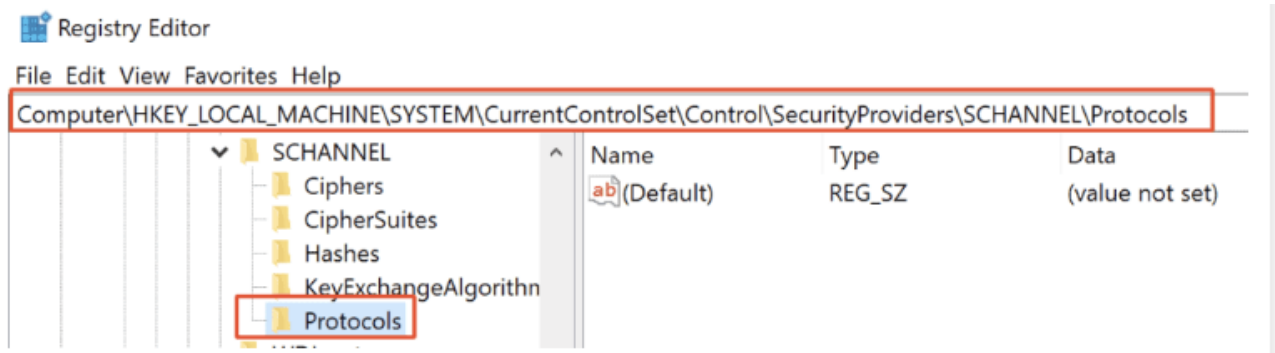


2. Run below commands to create Registry entry

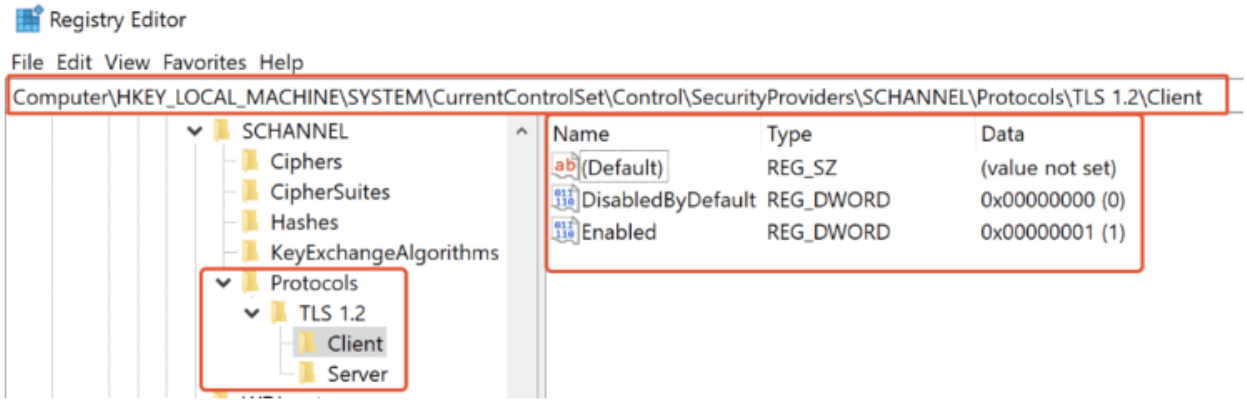
```
- New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Client' -Force

- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Client' -PropertyType 'DWORD' -Name 'DisabledByDefault' -Value '0'

- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Client' -PropertyType 'DWORD' -Name 'Enabled' -Value '1'



- New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Server' -Force

- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Server' -PropertyType 'DWORD' -Name 'DisabledByDefault' -Value '0'
```

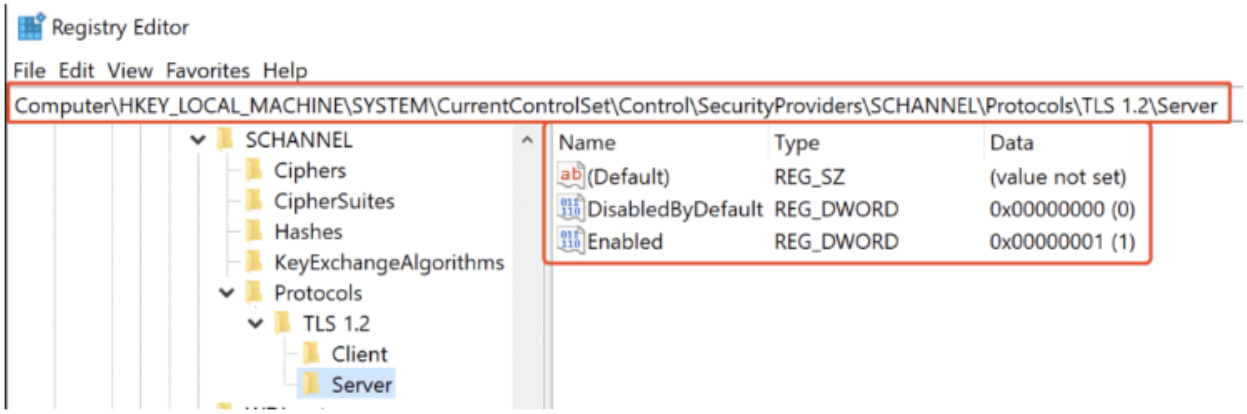
```
- New-ItemProperty -Path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS
1.2\Server' -PropertyType 'DWORD' -Name 'Enabled' -Value '1'



TLS 1.3 (Supports in Windows 11 & Windows Server 2022)
- New-ItemProperty -Path 'HKLM:\SYSTEM\CurrentControlSet\services\HTTP\Parameters' -
PropertyType 'DWORD' -Name 'EnableHttp3' -Value '1'
```

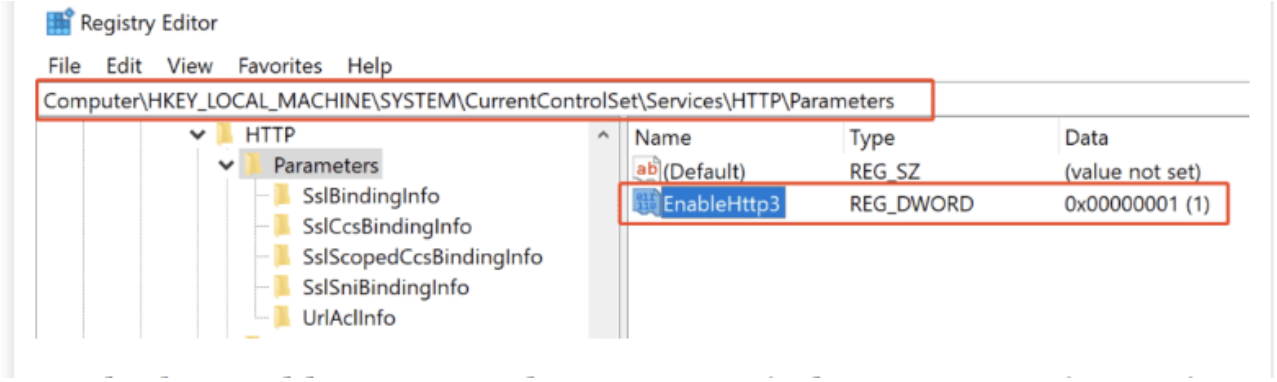
Before running the commands you can see no items were exist underneath Protocol.



After running the commands you can see there are two keys created 'TLS 1.2' & 'TLS 1.3', Underneath each protocols there are 'Client' &'Server' Keys inside them ther are two items 'DisableByDefault' & 'Enabled'.


Registry Editor
File Edit View Favorites Help
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Client
SCHANNEL
Ciphers
CipherSuites
Hashes
KeyExchangeAlgorithms
Protocols
TLS 1.2
Client
Server
Name
Type
Data
(Default)
REG_SZ
(value not set)
DisabledByDefault
REG_DWORD
0x00000000 (0)
Enabled
REG_DWORD
0x00000001 (1)

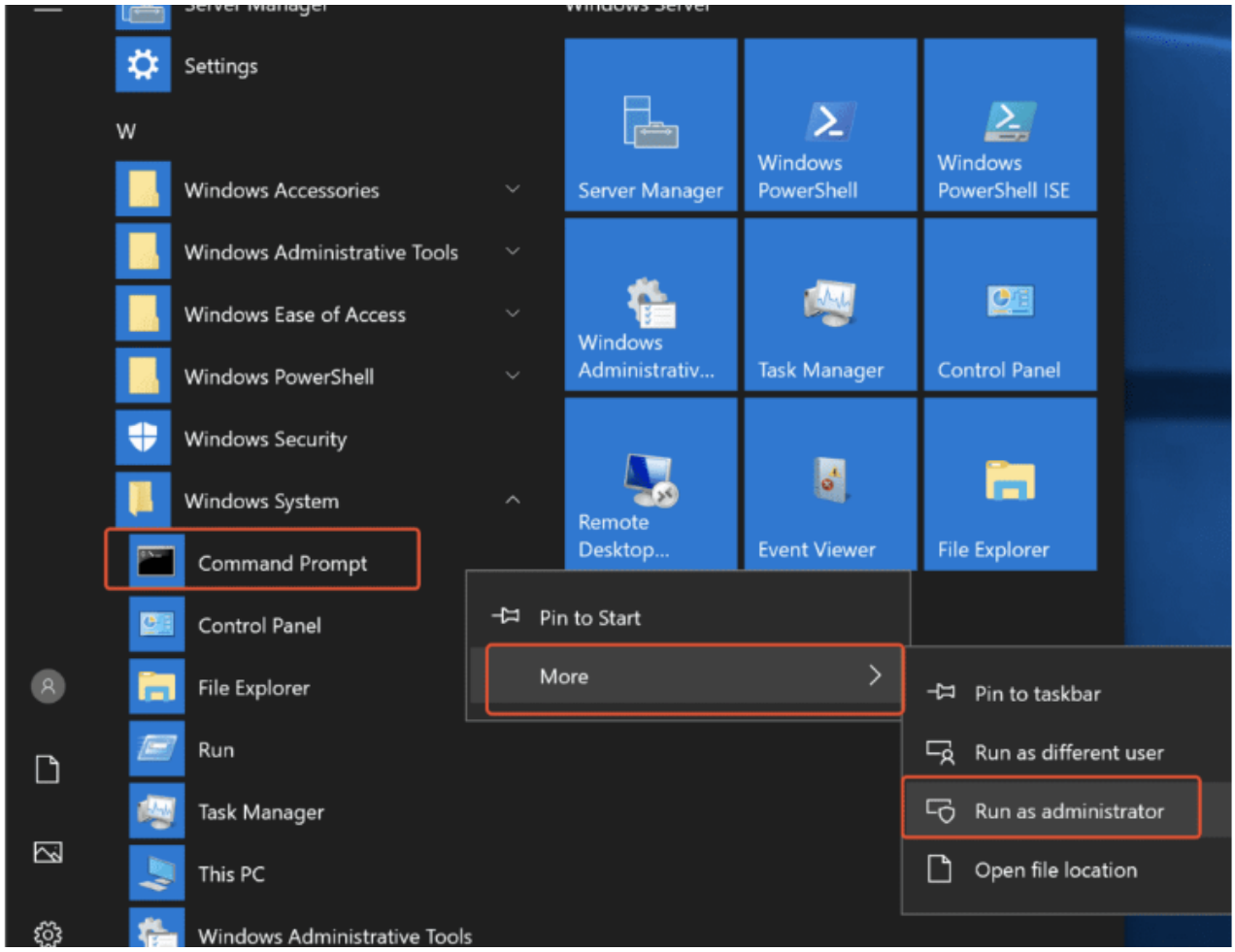

Registry Editor
File Edit View Favorites Help
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server
SCHANNEL
Ciphers
CipherSuites
Hashes
KeyExchangeAlgorithms
Protocols
TLS 1.2
Client
Server
Name
Type
Data
(Default)
REG_SZ
(value not set)
DisabledByDefault
REG_DWORD
0x00000000 (0)
Enabled
REG_DWORD
0x00000001 (1)


Registry Editor
File Edit View Favorites Help
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters
HTTP
Parameters
SslBindingInfo
SslCcsBindingInfo
SslScopedCcsBindingInfo
SslSniBindingInfo
UrlAclInfo
Name
Type
Data
(Default)
REG_SZ
(value not set)
EnableHttp3
REG_DWORD
0x00000001 (1)

Method 3: Enable TLS 1.2 and TLS 1.3 on Windows Server using native CMD

Follow this simple procedure to enable TLS 1.2 and TLS 1.2 using CMD comments.

1. Open ‘Command Prompt’ as Administrator.



2. Run below commands to create Registry entry.

```
- reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.2\Client" /v DisabledByDefault /t REG_DWORD /d 0 /f

- reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.2\Client" /v Enabled /t REG_DWORD /d 1 /f


- reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.2\Server" /v DisabledByDefault /t REG_DWORD /d 0 /f

- reg add
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\
TLS 1.2\Server" /v Enabled /t REG_DWORD /d 1 /f



TLS 1.3 (Supports in Windows 11 & Windows Server 2022)
- reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\HTTP\Parameters" /v
EnableHttp3 /t REG_DWORD /d 1 /f
```

Conclusion:

This paper proposed the Quantitative Information Security Risk As-sessment framework for E-commerce .The goal of the proposed model is to reduce risks of security breach, this means understanding the cause that makes .Applying the proposed framework onto a Popular E-commerce website .It's clear that the there our goal is to solve the security of a web-site browser and what kinds of vulnerability we can face and how can we solve those vulnerability by using different methods and tools .

The proposed model quantitatively measured the risk magnitude for E-commernce website network configuration and can be used by risk analyst and security manager of website e to perform reliable and repeatable risk anal-ysis in realistic and affordable manner.It's enables company to stay a step ahead of security threats and also to get more value from their security budget, by focusing on critical assets that are truly at risk.

Reference:

1.Stack Exchange :
https://security.stackexchange.com/questions/87283/is-tls-1-0-more-secure-than-tls-1-2

2.Microsoft :
https://support.microsoft.com/en-us/topic/update-to-enable-tls-1-1-and-tls-1-2-as-default-secure-protocols-in-winhttp-in-windows-c4bd73d2-31d7-761e-0178-11268bb10392

3.Acunetix

https://www.acunetix.com

4.https://scholar.google.com/scholar_lookup?title=Metasploit%3A%20the%20penetration%20tester%27s%20guide&author=D.%20Kennedy&publication_year=2011

5.Guide for applying the risk management framework to federal information sys-tems, U.S. Department of Commerce,

6. Prioritizing information security risks with threat agent risk assessment,

Whitepaper.

7. The secmaster :

https://thesecmaster.com/web-stories/enable-tls-1-3-on-web-servers/