TECHNICAL ANALYSIS

Mon April 28, 2025

Networks

A_AHS_Scan4_NoSIH

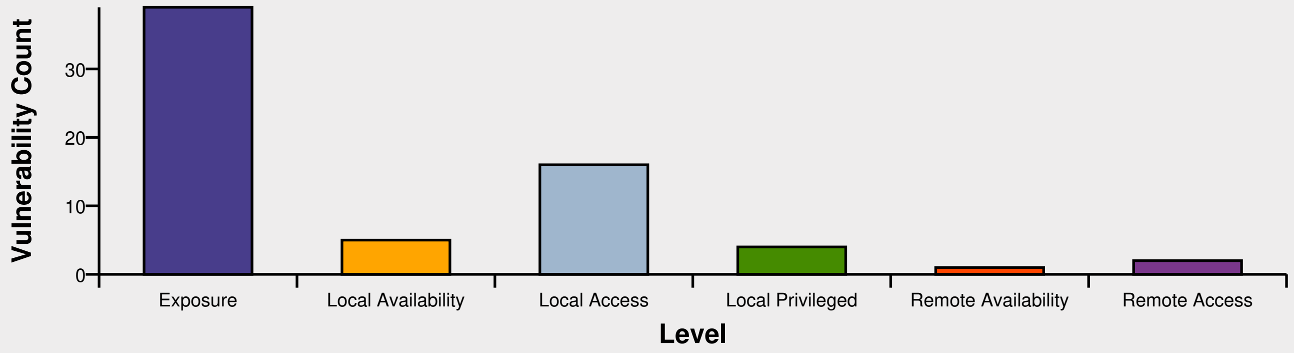
Filters

Windows OS Only

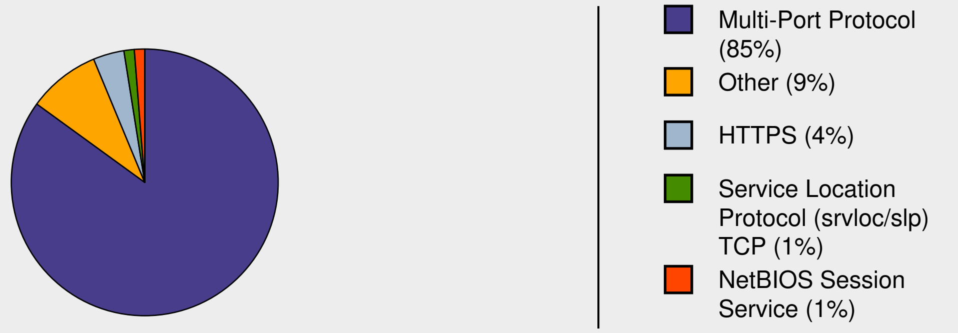
Report Summary

Networks/Network Groups	A_AHS_Scan4_NoSIH	**Filters**	Windows OS Only
Hosts	1	**Asset Value**	0
Average Host Score	195	**Vulnerabilities**	68
Applications/Services	81		

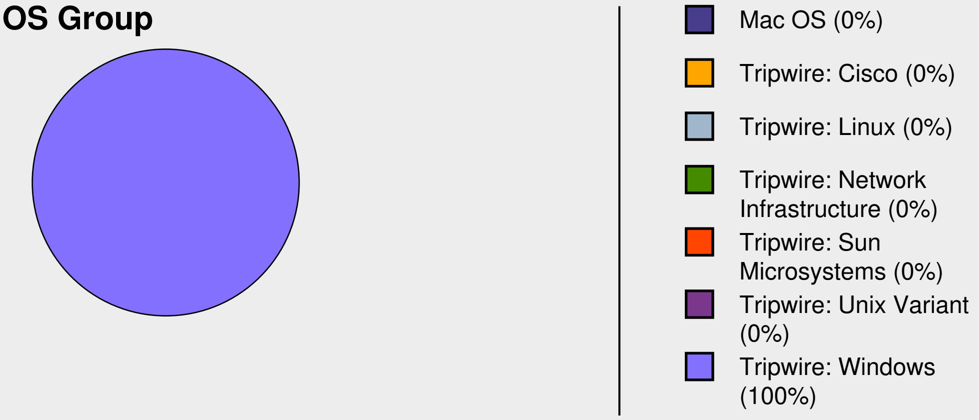
Vulnerability Level Distribution



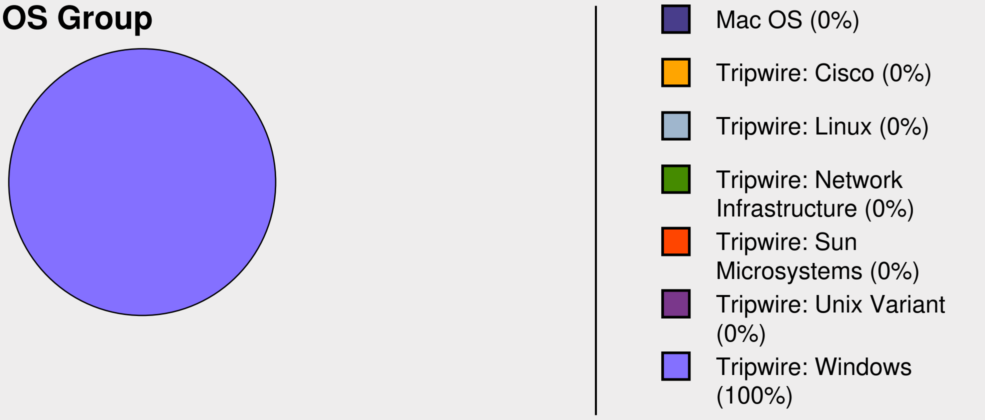
Service Distribution



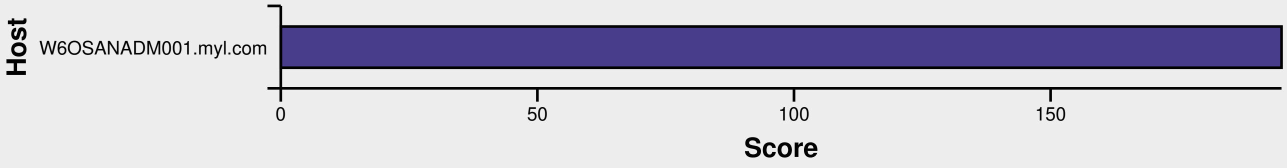
OS Distribution by OS Group



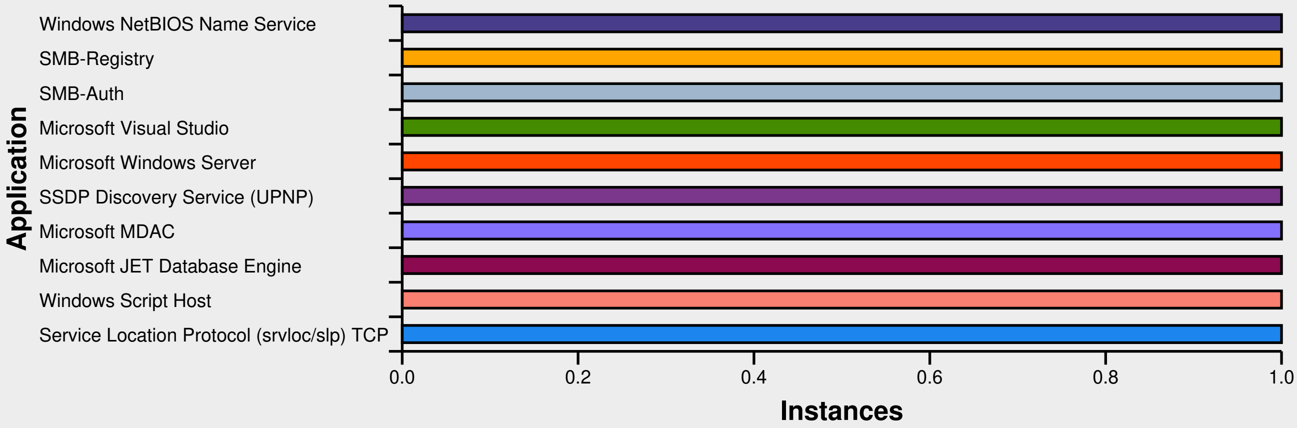
Vuln Distribution by OS Group

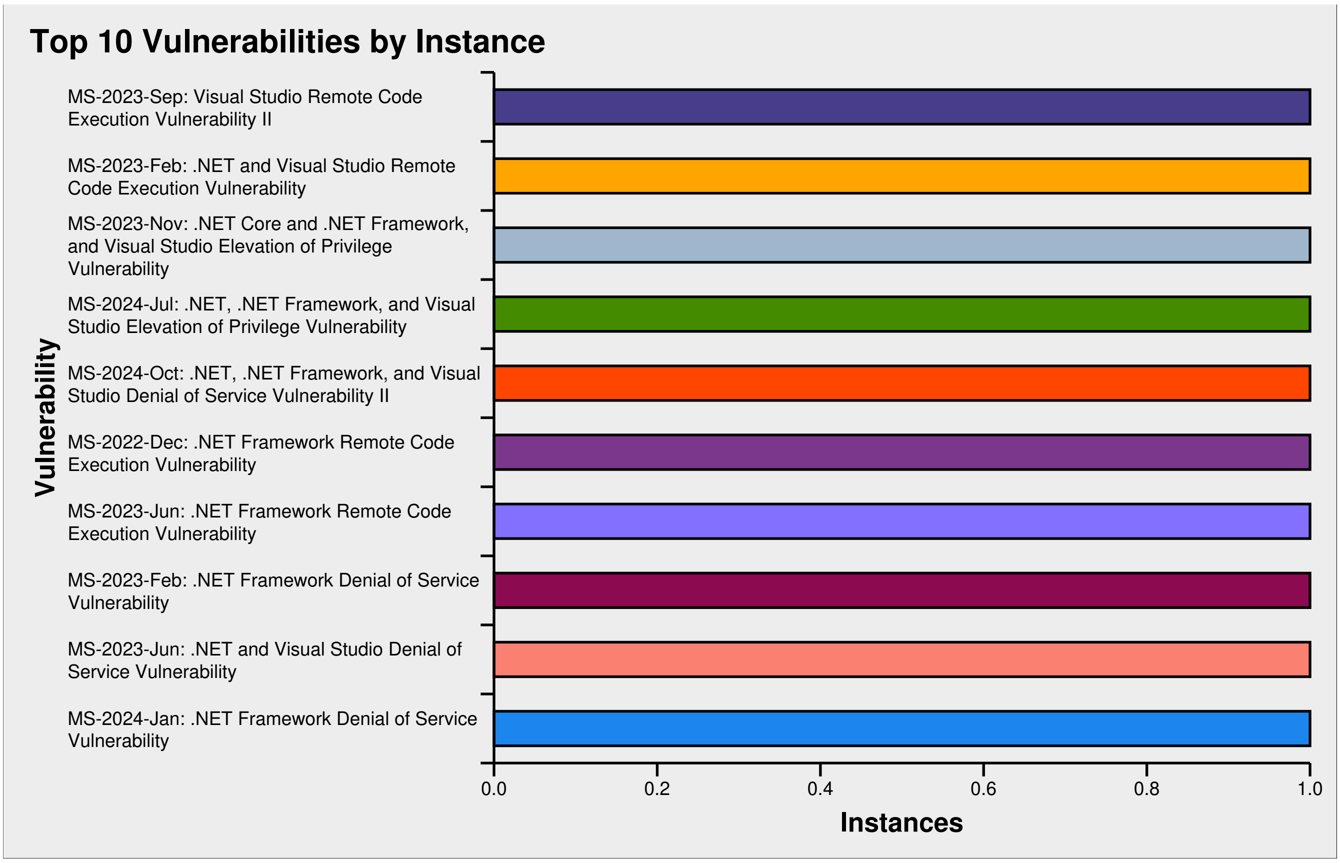


Top 10 Most Vulnerable Hosts



Top 10 Applications by Instance




Top 10 Vulnerabilities by Instance
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability II
MS-2023-Feb: .NET and Visual Studio Remote Code Execution Vulnerability
MS-2023-Nov: .NET Core and .NET Framework, and Visual Studio Elevation of Privilege Vulnerability
MS-2024-Jul: .NET, .NET Framework, and Visual Studio Elevation of Privilege Vulnerability
MS-2024-Oct: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability II
MS-2022-Dec: .NET Framework Remote Code Execution Vulnerability
MS-2023-Jun: .NET Framework Remote Code Execution Vulnerability
MS-2023-Feb: .NET Framework Denial of Service Vulnerability
MS-2023-Jun: .NET and Visual Studio Denial of Service Vulnerability
MS-2024-Jan: .NET Framework Denial of Service Vulnerability
Vulnerability
0.0
0.2
0.4
0.6
0.8
1.0
Instances

Hosts

Hostname	IP Address	OS	Agent	Owner	Asset Value	Score
W6OSANADM00	10.232.7.13	Windows Server 2019	No	None	0	195

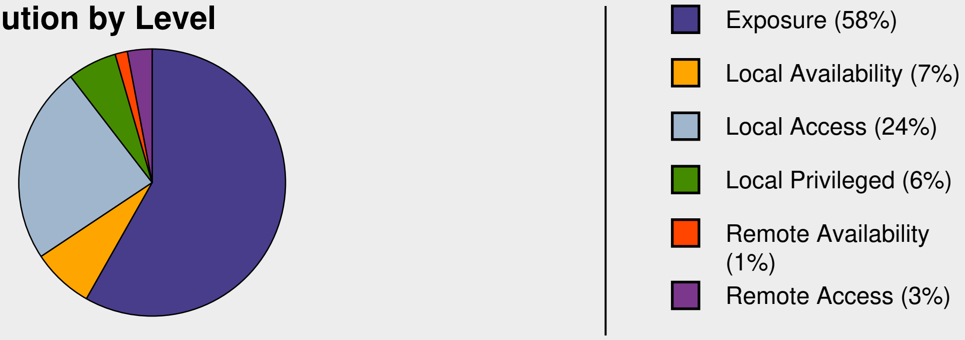
Host Summary

Hostname	W6OSANADM001.myl.com	**IP Address**	10.232.7.13
Score	195	**Asset Value**	0
OS Name	Windows Server 2019	**Owner**	None
NetBIOS Name	W6OSANADM001	**Mac Address (NetBIOS)**	
Domain/Workgroup	MYL		

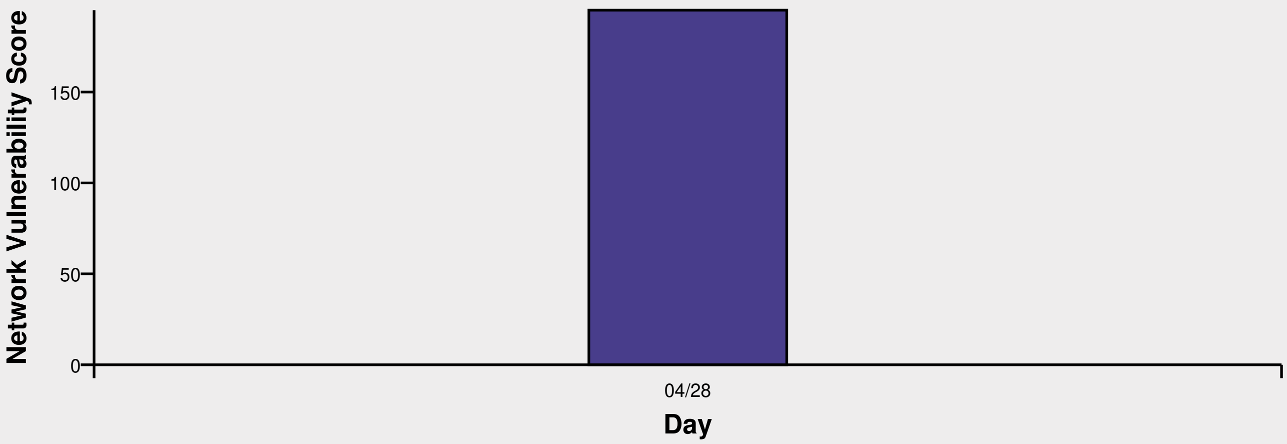
Operating System

OS Name
Windows Server 2019

Vulnerability Distribution by Level



Score Distribution by Day



Vulnerabilities

Vulnerability	CVE	# of Ports	Score
MS-2024-Jan: Microsoft.Data.SqlClient and System.Data.SqlClient SQL Data Provider Security Feature Bypass Vulnerability	CVE-2024-0056	1	72

continued on next page

Vulnerability	CVE	# of Ports	Score
MS-2024-Jan: NET, .NET Framework, and Visual Studio Security Feature Bypass Vulnerability	CVE-2024-0057	1	72
MS-2024-Jan: .NET Framework Denial of Service Vulnerability	CVE-2024-21312	1	14
MS-2019-Aug: Encryption Key Negotiation of Bluetooth Vulnerability	CVE-2019-9506	1	10
MS-2023-Aug: ASP.NET Elevation of Privilege Vulnerability	CVE-2023-36899	1	4
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	CVE-2023-24936	1	4
MS-2023-Nov: .NET Core and .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	CVE-2023-36049	1	3
MS-2024-Jul: .NET, .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	CVE-2024-38081	1	2
MS-2021-May: Microsoft Jet Red Database Engine Remote Code Execution Vulnerability	CVE-2021-28455	1	2
MS-2024-Apr: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability	CVE-2024-21409	1	1
MS-2022-Dec: .NET Framework Remote Code Execution Vulnerability	CVE-2022-41089	1	1
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability I	CVE-2023-24897	1	1
MS-2023-Jun: .NET Framework Remote Code Execution Vulnerability	CVE-2023-29326	1	1
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability II	CVE-2023-24895	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability I	CVE-2023-36792	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability II	CVE-2023-36796	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability III	CVE-2023-36794	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability IV	CVE-2023-36793	1	1
MS-2023-Sep: .NET Framework Remote Code Execution Vulnerability	CVE-2023-36788	1	1
MS-2023-Nov: ASP.NET Security Feature Bypass Vulnerability	CVE-2023-36560	1	1
MS-2023-Feb: .NET and Visual Studio Remote Code Execution Vulnerability	CVE-2023-21808	1	1
No UNC Paths Configured for Privacy		1	0
No UNC Paths Configured for Mutual Authentication		1	0
Windows DRT Command Success		1	0
RFC7525 Recommended Cipher Suites Exposure		1	0
MS15-124: Microsoft Browser ASLR Bypass Vulnerability	CVE-2015-6161	1	0
Perfect Forward Secrecy Preferred		1	0
Perfect Forward Secrecy Available		1	0
Google Chrome Enterprise Policy Site Isolation Per Process Not Enabled		1	0

continued on next page

Vulnerability	CVE	# of Ports	Score
TLSv1.2 Enabled		1	0
Remote Desktop Network Level Authentication (NLA) Enabled		1	0
CACHED APPLICATION DATA		1	0
DCE RPC mapper available		1	0
MIME Type Sniffing Disabled		1	0
ms-msdt Protocol Scheme Configured		1	0
search-ms Protocol Scheme Configured		1	0
Unquoted Service Path Weakness		1	0
MS-2022-Nov: .NET Framework Information Disclosure Vulnerability	CVE-2022-41064	1	0
MS-2023-Feb: .NET Framework Denial of Service Vulnerability	CVE-2023-21722	1	0
MS-2023-Jun: .NET and Visual Studio Denial of Service Vulnerability	CVE-2023-32030	1	0
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability	CVE-2023-29331	1	0
MS-2023-Aug: .NET Framework Spoofing Vulnerability	CVE-2023-36873	1	0
SSL Server Supports CBC Ciphers for TLSv1.2		1	0
MS-2024-Jul: Windows Cryptographic Services Security Feature Bypass Vulnerability	CVE-2024-30098	1	0
MS-2024-Oct: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability I	CVE-2024-43483	1	0
MS-2024-Oct: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability II	CVE-2024-43484	1	0
MS-2025-Jan: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability	CVE-2025-21176	1	0
X-XSS-Protection Enabled		1	0
HTTP Available		1	0
NetBIOS SSN Available		1	0
SMB AUTHENTICATION SUCCESS		1	0
Host has IPv6 Enabled		1	0
RPC DCOM AUTHENTICATION SUCCESS		1	0
WMI AUTHENTICATION SUCCESS		1	0
The contents of an SMB share may be enumerated		1	0
A Windows SMB share permits read access to Everyone [via SMB]		1	0
SSL/TLS Certificate Signature Validation Failed		1	0
Untrusted SSL/TLS Certificate		1	0
Microsoft Remote Desktop Service Available		1	0
IP Addresses Enumerated Via NetBIOS		1	0
Portable Storage Devices Detected (Windows)		1	0
SSL Certificate Information		1	0
UNRELIABLE SSL/TLS CERTIFICATE CHAIN		1	0
SSL Certificate Key Length < 4096 bits		1	0
SSL Certificate Key Length <= 2048 bits		1	0
SSL Certificate Key Length <= 4096 bits		1	0
BigFix		1	0
No UNC Paths Configured for Integrity		1	0

continued on next page

Vulnerability	CVE	# of Ports	Score

Applications

Service	Application	Port
DCE/MS RPC over TCP	DCE/MS RPC Endpoint Mapper Interface (TCP)	135
Direct SMB Hosting Service	Microsoft Windows OS Family 1809 Direct SMB Session Service	445
HTTPS	HTTP Server	8443
HTTPS	HTTP-Based Application	8443
HTTPS	TLSv1.2	8443
IPv4 Layer 4		0
Microsoft Remote Desktop Protocol	Windows 6.x-Windows 10.x (via RDP)	3389
Multi-Port Protocol	AllJoyn Router Service	0
Multi-Port Protocol	CNG Key Isolation Service	0
Multi-Port Protocol	DirectWrite	0
Multi-Port Protocol	DirectX 10.x	0
Multi-Port Protocol	DirectX 11 Build 17763	0
Multi-Port Protocol	DirectX 12 Build 17763	0
Multi-Port Protocol	DirectX 9.0c	0
Multi-Port Protocol	Google Chrome Extensions	0
Multi-Port Protocol	Google Chrome Versions	0
Multi-Port Protocol	HCL BigFix Client 10.0.7.52	0
Multi-Port Protocol	Host has IPv6 Enabled	0
Multi-Port Protocol	HTTP Service	0
Multi-Port Protocol	IKE and AuthIP IPsec Keying Modules Service	0
Multi-Port Protocol	Ink Support Feature	0
Multi-Port Protocol	IP Helper Service	0
Multi-Port Protocol	IPSec Policy Agent Service	0
Multi-Port Protocol	KDC Proxy Server Service	0
Multi-Port Protocol	Microsoft .NET Framework v4.7.x	0
Multi-Port Protocol	Microsoft Cryptographic Services	0
Multi-Port Protocol	Microsoft Internet Explorer 11	0
Multi-Port Protocol	Microsoft JET Database Engine	0
Multi-Port Protocol	Microsoft JScript	0
Multi-Port Protocol	Microsoft Korean Language IME	0
Multi-Port Protocol	Microsoft MDAC	0
Multi-Port Protocol	Microsoft Paint	0
Multi-Port Protocol	Microsoft Remote Desktop Protocol 10.0	0
Multi-Port Protocol	Microsoft SharePoint	0
Multi-Port Protocol	Microsoft SoftGrid/Application Virtualization	0
Multi-Port Protocol	Microsoft System Center Operations Monitoring Agent 2019	0
Multi-Port Protocol	Microsoft Terminal Services Client	0
Multi-Port Protocol	Microsoft VBScript	0
Multi-Port Protocol	Microsoft Visual Studio	0
Multi-Port Protocol	Microsoft Windows Server	0
Multi-Port Protocol	Microsoft Windows Telnet Client	0
Multi-Port Protocol	MPEG Layer-3 codecs	0
Multi-Port Protocol	MSXML 3.0	0

continued on next page

Service	Application	Port
Multi-Port Protocol	MSXML 6.0	0
Multi-Port Protocol	Print Spooler Service	0
Multi-Port Protocol	Remote Registry Service	0
Multi-Port Protocol	Smart Card Service	0
Multi-Port Protocol	SSDP Discovery Service (UPNP)	0
Multi-Port Protocol	Symantec AntiVirus	0
Multi-Port Protocol	Symantec Endpoint Protection Client	0
Multi-Port Protocol	Telephony Service	0
Multi-Port Protocol	USB Attached SCSI Protocol Service	0
Multi-Port Protocol	VMware Tools 12.4.5	0
Multi-Port Protocol	Volume Shadow Copy Service	0
Multi-Port Protocol	Windows Address Book	0
Multi-Port Protocol	Windows ATL Component	0
Multi-Port Protocol	Windows CloudExperienceHost Broker	0
Multi-Port Protocol	Windows Core Messaging	0
Multi-Port Protocol	Windows Domain Joined Host	0
Multi-Port Protocol	Windows Mail	0
Multi-Port Protocol	Windows Media Player 12	0
Multi-Port Protocol	Windows OpenSSH Client	0
Multi-Port Protocol	Windows OS (Not Server Core)	0
Multi-Port Protocol	Windows Projected File System	0
Multi-Port Protocol	Windows Remote Access Connection Manager	0
Multi-Port Protocol	Windows Remote Desktop Available	0
Multi-Port Protocol	Windows Remote Desktop Configuration Service	0
Multi-Port Protocol	Windows Script Host	0
Multi-Port Protocol	Windows Search / Windows Desktop Search	0
Multi-Port Protocol	Windows Secure Boot Enabled	0
Multi-Port Protocol	Windows Server 2019	0
Multi-Port Protocol	Windows Workstation Service	0
Multi-Port Protocol	WinSCP 6.x	0
Multi-Port Protocol	Wireless LAN AutoConfig Service Running	0
Multi-Port Protocol	WordPad	0
NetBIOS Name Service	Windows NetBIOS Name Service	137
NetBIOS Session Service	Microsoft Windows OS Family 1809 NetBIOS Session Service	139
Open TCP Port	N/A	8088
Service Location Protocol (srvloc/slp) TCP		427
SMB-Auth	N/A	0
SMB-Registry	N/A	0

Configuration Checks

Configuration Check	Discovery Method	Value
All Hardened UNC Paths Found	WDRT	{}
AllowEncryptionOracle	WDRT	AllowEncryptionOracle registry reports force updated clients mode (0x00000000).
Automatic Updates Enabled	WDRT	Windows version does not support Automatic Updates

continued on next page

Configuration Check	Discovery Method	Value
Bad Certificate Chain	SSL	The following problems have been detected for the certificate chain provided by service on TCP(8443): [Certificate: E6:31:2E:A5:66:B9:DC:3C:93:8E:8C:45:F2:4B:7B:0C:4D:48:DA:9E:7A:00:94:14:FB:D retrieved with hostnames: <NO SNI>]: One or more certificates in the chain is unsupported for verification.
DNS Computer Name	TCP	TCP(139): W6OSANADM001.myl.com, TCP(445): W6OSANADM001.myl.com
DNS Domain Name	TCP	TCP(139): myl.com, TCP(445): myl.com
DNS Tree Name	TCP	TCP(139): myl.com, TCP(445): myl.com
Google Chrome Version	WDRT	135.0.7049.115
HTTP Supported Methods	TCP	GET, HEAD
IP Addresses via NETBIOS	UDP	10.232.7.13
Last Logged In User	WDRT	MYL\M677261_sadm
Netbios Computer Name	TCP	TCP(139): W6OSANADM001, TCP(445): W6OSANADM001
Netbios Domain Name	TCP	TCP(139): MYL, TCP(445): MYL
Nmap OS String	TCP	
Nmap Status	NMAP	Global: Nmap Not Configured
SMB Shares Everyone File System Read Access	SMB	D$, P$
SMB Shares Where Contents May Be Enumerated	SMB	ADMIN$, C$, D$, GEO_DRIVE, O$, P$
SMB Username	SMB	myl\\svc_ncirclecred
SSL Certificate Extended Key Usage	SSL	TCP(3389): serverAuth , TCP(8443):
SSL Certificate Issuer	SSL	TCP(3389): commonName=W6OSANADM001.myl.com, TCP(8443): organizationalUnitName=CTD\, organizationName=EMC\, stateName=MA\, commonName=W6OSANADM001.myl.com\, countryName=US\, localityName=HOPKINTON
SSL Certificate Key Usage	SSL	TCP(3389): keyEncipherment dataEncipherment , TCP(8443):
SSL Certificate MD5 Thumbprint	SSL	TCP(3389): 69:0C:69:E7:00:C0:0A:B9:95:9A:57:B8:61:B0:FC:AF, TCP(8443): B6:60:81:7A:C3:B6:2C:35:64:D5:7E:8C:74:B8:85:E1
SSL Certificate Public Key Size	SSL	TCP(3389): 2048 bits, TCP(8443): 2048 bits
SSL Certificate SHA1 Thumbprint	SSL	TCP(3389): AA:3C:98:61:D1:BD:A8:4C:4A:A3:F2:ED:19:26:13:CB:79:C6:18:CD, TCP(8443): C6:4F:F5:D5:D9:09:31:B3:75:6C:9C:2A:0C:4B:2B:37:71:72:78:61
SSL Certificate Serial Number	SSL	TCP(3389): 18:9D:FB:98:D0:8C:B6:83:4C:7A:1C:62:51:D0:95:4B, TCP(8443): 30:0C:FF:C2
SSL Certificate Signature Algorithm	SSL	TCP(3389): sha256WithRSAEncryption, TCP(8443): Not Available
SSL Certificate Subject	SSL	TCP(3389): commonName=W6OSANADM001.myl.com, TCP(8443): organizationalUnitName=CTD\, organizationName=EMC\, stateName=MA\, commonName=W6OSANADM001.myl.com\, countryName=US\, localityName=HOPKINTON

continued on next page

Configuration Check	Discovery Method	Value
SSL Certificate Valid From	SSL	TCP(3389): Thu Apr 17 08:00:13 2025 UTC, TCP(8443): Fri May 27 13:58:26 2022 UTC
SSL Certificate Valid To	SSL	TCP(3389): Fri Oct 17 08:00:13 2025 UTC, TCP(8443): Mon May 24 13:58:26 2032 UTC
SSL/TLS Enabled Ciphers	SSL	TCP(3389) TLSv1.2: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384\, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256\, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384\, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256\, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA\, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA\, TLS_RSA_WITH_AES_256_GCM_SHA384\, TLS_RSA_WITH_AES_128_GCM_SHA256\, TLS_RSA_WITH_AES_256_CBC_SHA256\, TLS_RSA_WITH_AES_128_CBC_SHA256\, TLS_RSA_WITH_AES_256_CBC_SHA\, TLS_RSA_WITH_AES_128_CBC_SHA; , TCP(8443) TLSv1.2: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256\, TLS_RSA_WITH_AES_128_CBC_SHA256\, TLS_RSA_WITH_AES_256_CBC_SHA256\, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384\, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384\, TLS_RSA_WITH_AES_128_GCM_SHA256\, TLS_RSA_WITH_AES_256_GCM_SHA384\, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256;
Secure Authentication Sequence Required for Logon	SMB	1
TLSv1.2 CBC Ciphers	SSL	TCP(8443) TLSv1.2: TLS_RSA_WITH_AES_128_CBC_SHA256\, TLS_RSA_WITH_AES_256_CBC_SHA256\, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384\, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLSv1.2 Strong Ciphers	SSL	TCP(3389): TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (128-bit)\, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (256-bit)\, TLS_RSA_WITH_AES_128_GCM_SHA256 (128-bit)\, TLS_RSA_WITH_AES_256_GCM_SHA384 (256-bit), TCP(8443): TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (128-bit)\, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (256-bit)\, TLS_RSA_WITH_AES_256_GCM_SHA384 (256-bit)\, TLS_RSA_WITH_AES_128_GCM_SHA256 (128-bit)
USB Devices Detected on Windows	SMB	Unnamed Devices: ['@usbhub3.inf\,%usbhub3.roothubdevicedesc%;USB Root Hub (USB 3.0)'\, '@usb.inf\,%usb\\\\composite.devicedesc%;USB Composite Device'\, '@input.inf\,%hid.devicedesc%;USB Input Device'\, '@input.inf\,%hid.devicedesc%;USB Input Device'\, '@usb.inf\,%usb\\\\composite.devicedesc%;USB Composite Device'\, '@input.inf\,%hid.devicedesc%;USB Input Device']

continued on next page

Configuration Check	Discovery Method	Value
Unquoted Service Paths	WDRT	BHDrvx64: \??\C:\ProgramData\Symantec\Symantec Endpoint Protection\14.3.8289.5000.105\Data\Definitions\BASHDefs\20250424.001\BHDrvx Symantec Eraser Control driver: \??\C:\Program Files (x86)\Common Files\Symantec Shared\EENGINE\eeCtrl64.sys, EraserUtilRebootDrv: \??\C:\Program Files (x86)\Common Files\Symantec Shared\EENGINE\EraserUtilRebootDrv.sys, IDSVia64: \??\C:\ProgramData\Symantec\Symantec Endpoint Protection\14.3.8289.5000.105\Data\Definitions\IPSDefs\20250425.061\IDSvia64.sys, Symantec Real Time Storage Protection x64: \??\C:\ProgramData\Symantec\Symantec Endpoint Protection\14.3.8289.5000.105\Data\SymPlatform\SRTSP64.SYS, Symantec Eventing Platform: \??\C:\ProgramData\Symantec\Symantec Endpoint Protection\14.3.8289.5000.105\Data\SymPlatform\SymEvnt.sys
WDRT Authentication Success	TCP	True
WDRT Protocol Used	WDRT	SMB Registry and File Access, 64-bit
WDRT_Access	TCP	WDRT_SMB_AUTH_SUCCESS : True, WDRT_SMB_REGISTRY_ACCESS : True, WDRT_SMB_FILE_ACCESS : True, WDRT_RPC_AUTH_SUCCESS : True, WDRT_WMI_AUTH_SUCCESS : True, WDRT_HOST_IS_64BIT : True,
Windows Build Version	WDRT	17763.7249
Windows DRT Access	WDRT	Windows Registry Access: True, CIFS Filesystem Access: True
Windows Edition	WDRT	Windows Server 2019 Standard
Windows IPv6 Setting	WDRT	DisabledComponents registry key is not present. All IPv6 components are enabled.
Windows Installer Version	WDRT	5.0.17763
Windows System Root Directory	SMB	C:\Windows

Vulnerabilities

Vulnerability	CVE	Hosts	Score
MS-2024-Jan: Microsoft.Data.SqlClient and System.Data.SqlClient SQL Data Provider Security Feature Bypass Vulnerability	CVE-2024-0056	1	72
MS-2024-Jan: NET, .NET Framework, and Visual Studio Security Feature Bypass Vulnerability	CVE-2024-0057	1	72
MS-2024-Jan: .NET Framework Denial of Service Vulnerability	CVE-2024-21312	1	14
MS-2019-Aug: Encryption Key Negotiation of Bluetooth Vulnerability	CVE-2019-9506	1	10
MS-2023-Aug: ASP.NET Elevation of Privilege Vulnerability	CVE-2023-36899	1	4
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	CVE-2023-24936	1	4
MS-2023-Nov: .NET Core and .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	CVE-2023-36049	1	3
MS-2024-Jul: .NET, .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	CVE-2024-38081	1	2
MS-2021-May: Microsoft Jet Red Database Engine Remote Code Execution Vulnerability	CVE-2021-28455	1	2
MS-2024-Apr: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability	CVE-2024-21409	1	1
MS-2022-Dec: .NET Framework Remote Code Execution Vulnerability	CVE-2022-41089	1	1
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability I	CVE-2023-24897	1	1
MS-2023-Jun: .NET Framework Remote Code Execution Vulnerability	CVE-2023-29326	1	1
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability II	CVE-2023-24895	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability I	CVE-2023-36792	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability II	CVE-2023-36796	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability III	CVE-2023-36794	1	1
MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability IV	CVE-2023-36793	1	1
MS-2023-Sep: .NET Framework Remote Code Execution Vulnerability	CVE-2023-36788	1	1
MS-2023-Nov: ASP.NET Security Feature Bypass Vulnerability	CVE-2023-36560	1	1
MS-2023-Feb: .NET and Visual Studio Remote Code Execution Vulnerability	CVE-2023-21808	1	1
No UNC Paths Configured for Privacy		1	0
No UNC Paths Configured for Mutual Authentication		1	0
Windows DRT Command Success		1	0
RFC7525 Recommended Cipher Suites Exposure		1	0

continued on next page

Vulnerability	**CVE**	**Hosts**	**Score**
MS15-124: Microsoft Browser ASLR Bypass Vulnerability	CVE-2015-6161	1	0
Perfect Forward Secrecy Preferred		1	0
Perfect Forward Secrecy Available		1	0
Google Chrome Enterprise Policy Site Isolation Per Process Not Enabled		1	0
TLSv1.2 Enabled		1	0
Remote Desktop Network Level Authentication (NLA) Enabled		1	0
CACHED APPLICATION DATA		1	0
DCE RPC mapper available		1	0
MIME Type Sniffing Disabled		1	0
ms-msdt Protocol Scheme Configured		1	0
search-ms Protocol Scheme Configured		1	0
Unquoted Service Path Weakness		1	0
MS-2022-Nov: .NET Framework Information Disclosure Vulnerability	CVE-2022-41064	1	0
MS-2023-Feb: .NET Framework Denial of Service Vulnerability	CVE-2023-21722	1	0
MS-2023-Jun: .NET and Visual Studio Denial of Service Vulnerability	CVE-2023-32030	1	0
MS-2023-Jun: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability	CVE-2023-29331	1	0
MS-2023-Aug: .NET Framework Spoofing Vulnerability	CVE-2023-36873	1	0
SSL Server Supports CBC Ciphers for TLSv1.2		1	0
MS-2024-Jul: Windows Cryptographic Services Security Feature Bypass Vulnerability	CVE-2024-30098	1	0
MS-2024-Oct: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability I	CVE-2024-43483	1	0
MS-2024-Oct: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability II	CVE-2024-43484	1	0
MS-2025-Jan: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability	CVE-2025-21176	1	0
X-XSS-Protection Enabled		1	0
HTTP Available		1	0
NetBIOS SSN Available		1	0
SMB AUTHENTICATION SUCCESS		1	0
Host has IPv6 Enabled		1	0
RPC DCOM AUTHENTICATION SUCCESS		1	0
WMI AUTHENTICATION SUCCESS		1	0
The contents of an SMB share may be enumerated		1	0
A Windows SMB share permits read access to Everyone [via SMB]		1	0
SSL/TLS Certificate Signature Validation Failed		1	0
Untrusted SSL/TLS Certificate		1	0
Microsoft Remote Desktop Service Available		1	0
IP Addresses Enumerated Via NetBIOS		1	0
Portable Storage Devices Detected (Windows)		1	0
SSL Certificate Information		1	0
UNRELIABLE SSL/TLS CERTIFICATE CHAIN		1	0
SSL Certificate Key Length < 4096 bits		1	0

continued on next page

Vulnerability	CVE	Hosts	Score
SSL Certificate Key Length <= 2048 bits		1	0
SSL Certificate Key Length <= 4096 bits		1	0
BigFix		1	0
No UNC Paths Configured for Integrity		1	0

Vulnerability

Vulnerability Name	MS-2024-Jan: Microsoft.Data.SqlClient and System.Data.SqlClient SQL Data Provider Security Feature Bypass Vulnerability	**Score**	72
Published	2024-01-09	**Strategy**	Data-Driven Attack
	nCircle: 600817	**CVSS v2**	4.0
CVSS v3	8.7		

Description

DESCRIPTION
Microsoft .NET Framework and Microsoft SQL Server is subject to a security feature bypass vulnerability. A local attacker could bypass security checks upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft SQL Server 2022
Microsoft Visual Studio 2022
Windows Registry

Advisory Publisher Entries

CVE:CVE-2024-0056	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-0056
CVSSv3 Base Score: 8.7	http://www.tripwire.com/vert/cvss/?data=8.7
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/
CWE: 319	http://cwe.mitre.org/data/definitions/319.html
MSRC Guidance: CVE-2024-0056	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-0056
Tripwire CVSSv3 Temporal Score: 4.7	http://www.tripwire.com/vert/cvss/?data=4.7
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)

continued on next page

Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1088 on 2024-01-10	http://www.tripwire.com/vert/?Released in ASPL 1088 on 2024-01-10

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll
", startVersion="2.0.50727", patchedVersion="2.0.50727.9063" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302.1" ) THEN
CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll", startVersion="2.0.50727", patchedVersion
="2.0.50727.9176" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302.1" ) THEN
CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion="4.8.
9214.0" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll", st
artVersion="2.0.50727", patchedVersion="2.0.50727.8976" )
CALL isOSFamily( osFamily="6.1,6.2,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="
4.7", fileName="system.dll", startVersion="4.7", patchedVersion="4.7.4081.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" )
THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion=
"4.8.4690.0" )
CALL isOSFamily( osFamily="6.3,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileNam
e="system.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8976" )
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.26'):  rule.STOP(True) elif V('7.0') <= ver < V('
7.0.15'):  rule.STOP(True) elif V('8.0') <= ver < V('8.0.1'):  rule.STOP(True)
rule.STOP(Fa
lse) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file='instapi160.dll'):  rule.RegistryGetValue(path) if not rule.success:  rule.STOP(False)
try:  path = r'%sShared\%s' % (rule.buffer,file) file_ver = smb_file.GetFileVersion(rule,
None, path) print file_ver ver = V(None, None, file_ver) except VE: rule.STOP(Fals
e) return ver
path = r'HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\160\VerSpecificRootDir'
if V('20
22') <= get_file_version(path) < V('2022.160.1110.1'):  rule.STOP(True) elif V('2022.160.4003') <= get_file
_version(path) < V('2022.160.4100.1'):  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

continued on next page

Hostname	IP Address	Score

Vulnerability

Vulnerability Name	MS-2024-Jan: NET, .NET Framework, and Visual Studio Security Feature Bypass Vulnerability	**Score**	72
Published	2024-01-09	**Strategy**	Data-Driven Attack
	nCircle: 600796	**CVSS v2**	4.0
CVSS v3	9.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a security feature bypass vulnerability. A remote attacker could bypass security checks upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2024-0057	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-0057
CVSSv3 Base Score: 9.8	http://www.tripwire.com/vert/cvss/?data=9.8
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I
CWE: 20	http://cwe.mitre.org/data/definitions/20.html
MSRC Guidance: CVE-2024-0057	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-0057
Tripwire CVSSv3 Temporal Score: 4.7	http://www.tripwire.com/vert/cvss/?data=4.7
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)

continued on next page

Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1088 on 2024-01-10	http://www.tripwire.com/vert/?Released in ASPL 1088 on 2024-01-10

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll
", startVersion="2.0.50727", patchedVersion="2.0.50727.9063" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302.1" ) THEN
CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll", startVersion="2.0.50727", patchedVersion
="2.0.50727.9176" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302.1" ) THEN
CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion="4.8.
9214.0" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll", st
artVersion="2.0.50727", patchedVersion="2.0.50727.8976" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll", st
artVersion="2.0.50727", patchedVersion="2.0.50727.8976" )
CALL isOSFamily( osFamily="6.1,6.2,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="
4.7", fileName="system.dll", startVersion="4.7", patchedVersion="4.7.4081.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" )
THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion=
"4.8.4690.0" )
CALL isOSFamily( osFamily="6.3,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileNam
e="system.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8976" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable(’PowerShell_Core_Version’) ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V(’7.0’) <= ver < V(’7.2.18’):  rule.STOP(True) elif V(’7.3’) <= ver < V(’7.3.11’):  rule.STOP(Tr
ue) elif V(’7.4’) <= ver < V(’7.4.2’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.26’):  rule.STOP(True) elif V(’7.0’) <= ver < V(’
7.0.15’):  rule.STOP(True) elif V(’8.0’) <= ver < V(’8.0.1’):  rule.STOP(True)
rule.STOP(Fa
lse) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2024-Jan: .NET Framework Denial of Service Vulnerability	**Score**	14
Published	2024-01-09	**Strategy**	DoS
	nCircle: 600826	**CVSS v2**	5.4
CVSS v3	7.5		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a denial of service vulnerability. An attacker could cause a denial of service condition upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2024-21312	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-21312
CVSSv3 Base Score: 7.5	http://www.tripwire.com/vert/cvss/?data=7.5
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I
CWE: 20	http://cwe.mitre.org/data/definitions/20.html
MSRC Guidance: CVE-2024-21312	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-21312
Tripwire CVSSv3 Temporal Score: 5.9	http://www.tripwire.com/vert/cvss/?data=5.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1088 on 2024-01-10	http://www.tripwire.com/vert/?Released in ASPL 1088 on 2024-01-10

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll
", startVersion="2.0.50727", patchedVersion="2.0.50727.9063" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0" ) THEN CALL isDotN
etVulnerable( dotNetVersion="2.0", fileName="system.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.
```

9176")
CALL isOSFamily(osFamily="10.0.2102,10.0.2202,11.0.2102,10.0.2102.1,11.0.2202,11.0.2302.0") THEN CALL isDotNetVulnerable(dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion="4.8.9214.0")
CALL isOSFamily(osFamily="6.1,6.2,10.0.1.0,10.0.0.2,10.0.0.6") THEN CALL isDotNetVulnerable(dotNetVersion="4.7", fileName="system.dll", startVersion="4.7", patchedVersion="4.7.4081.0")
CALL isOSFamily(osFamily="6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102") THEN CALL isDotNetVulnerable(dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion="4.8.4690.0")
CALL isOSFamily(osFamily="6.2,6.3,10.0.1.0,10.0.0.2") THEN CALL isDotNetVulnerable(dotNetVersion="2.0", fileName="system.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8976")

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2019-Aug: Encryption Key Negotiation of Bluetooth Vulnerability	**Score**	10
Published	2019-08-13	**Strategy**	Data-Driven Attack
	nCircle: 427755	**CVSS v2**	4.8
CVSS v3	8.1		

Description

DESCRIPTION
Bluetooth BR/EDR (aka Bluetooth Classic) contains a key negotiation vulnerability. An attacker with specialized hardware and in close proximity to the Bluetooth device could use this vulnerability to negotiate a key length with one byte of entropy.
SOLUTION
The vendor has released a software update for this vulnerability. A registry key must be configured after installing the update to resolve this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

CVE:CVE-2019-9506	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-9506
CVSSv3 Base Score: 8.1	http://www.tripwire.com/vert/cvss/?data=8.1
CVSSv3 Base Vector: CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:A/AC:L/PR:N/UI:N/S:U/C:H/I
CWE: 310	http://cwe.mitre.org/data/definitions/310.html
MSRC Guidance: CVE-2019-9506	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-9506
Tripwire CVSSv3 Temporal Score: 9.1	http://www.tripwire.com/vert/cvss/?data=9.1
Tripwire CVSSv3 Temporal Vector: (E:F/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:F/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 845 on 2019-08-14	http://www.tripwire.com/vert/?Released in ASPL 845 on 2019-08-14

Rules

```
EXECUTE { import smb_file from version import Version as V, VersionException as VE
rule.RegistryGetValue( r'H
KLM\System\CurrentControlSet\Policies\Hardware\Bluetooth\EnableMinimumEncryptionKeySize') if not rule.success
or rule.buffer != '0x00000001': rule.STOP(True)
try: win_ver = env.getHostVariable( 'windows_version'
) except KeyError:  rule.STOP( False )
def get_file_version( path, file=r’system32\\ntoskrnl.exe’ ):
try:  path = r’%s\\%s’ % (path,file) file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
try:  path =
env.getHostVariable(’windows_system_root_directory’) except KeyError:  rule.STOP(False)
if win_ver.start
swith( ’10.0.0.0’ ) and V( ’10.0’ ) <= get_file_version( path ) < V( ’10.0.10240.18303’ ):  rule.STOP(True)
elif win_ver.startswith( ’10.0.0.2’ ) and V( ’10.0’ ) <= get_file_version( path ) < V( ’10.0.14393.3143’ ):
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Aug: ASP.NET Elevation of Privilege Vulnerability	**Score**	4
Published	2023-08-08 nCircle: 585578	**Strategy** **CVSS v2**	Data-Driven Attack 5.5
CVSS v3	8.8		

Description

DESCRIPTION
Microsft .Net Framework is subject to a elevation of privilege vulnerability. A local attacker could elevate privileges upon successful exploitation of this vulnerability
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36899	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36899
CVSSv3 Base Score: 8.8	http://www.tripwire.com/vert/cvss/?data=8.8
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I
CWE: 20	http://cwe.mitre.org/data/definitions/20.html
MSRC Guidance: CVE-2023-36899	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36899
Tripwire CVSSv3 Temporal Score: 7.1	http://www.tripwire.com/vert/cvss/?data=7.1
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1068 on 2023-08-09	http://www.tripwire.com/vert/?Released in ASPL 1068 on 2023-08-09

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9061" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
```

```
Vulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9176.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9174" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", file
Name="system.web.dll", startVersion="4.7", patchedVersion="4.7.4057.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11
.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", p
atchedVersion="4.8.4654.0" )
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Jun: .NET, .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	**Score**	4
Published	2023-06-13 nCircle: 581548	**Strategy** **CVSS v2**	Data-Driven Attack 5.5
CVSS v3	7.5		

Description

DESCRIPTION
Microsoft .NET Framework and Visual Studios are subject to a elevation of privilege vulnerability. A local attacker could elevate privileges upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-24936	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-24936
CVSSv3 Base Score: 7.5	http://www.tripwire.com/vert/cvss/?data=7.5
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:H/
MSRC Guidance: CVE-2023-24936	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-24936
Tripwire CVSSv3 Temporal Score: 7.1	http://www.tripwire.com/vert/cvss/?data=7.1
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)

continued on next page

Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1060 on 2023-06-14	http://www.tripwire.com/vert/?Released in ASPL 1060 on 2023-06-14

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="mscorlib.d
ll", startVersion="4.0.30319", patchedVersion="4.6.1912.0" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9058" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9166.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9171" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4050.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11
.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", pat
chedVersion="4.8.4644.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable(’PowerShell_Core_Version’) ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V(’7.2’) <= ver < V(’7.2.12’):  rule.STOP(True) if V(’7.3’) <= ver < V(’7.3.5’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.18’):  rule.STOP(True) elif V(’7.0’) <= ver < V(’
7.0.7’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Nov: .NET Core and .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	**Score**	3
Published	2023-11-14	**Strategy**	Data-Driven Attack
	nCircle: 594058	**CVSS v2**	5.5
CVSS v3	9.8		

Description

DESCRIPTION
Microsft .Net Framework is subject to a elevation of privilege vulnerability. A local attacker could elevate privileges upon successful exploitation of this vulnerability
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36049	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36049
CVSSv3 Base Score: 9.8	http://www.tripwire.com/vert/cvss/?data=9.8
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I
CWE: 20	http://cwe.mitre.org/data/definitions/20.html
MSRC Guidance: CVE-2023-36049	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36049
Tripwire CVSSv3 Temporal Score: 7.1	http://www.tripwire.com/vert/cvss/?data=7.1
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)

continued on next page

Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1082 on 2023-11-15	http://www.tripwire.com/vert/?Released in ASPL 1082 on 2023-11-15

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="system.web
.dll", startVersion="4.0.30319", patchedVersion="4.6.1929.0" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9062" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302
.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patche
dVersion="4.8.9206.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302.1" ) THEN
CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVer
sion="2.0.50727.9175" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8975" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8975" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8975" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11
.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", p
atchedVersion="4.8.4682.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="4.7", fileName="system.web.dll", startVersion="4.7", patchedVersion="4.7.4076.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable('PowerShell_Core_Version') ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V('7.0') <= ver < V('7.2.17'):  rule.STOP(True) elif V('7.3') <= ver < V('7.3.10'):  rule.STOP(Tr
ue)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.25'):  rule.STOP(True) elif V('7.0') <= ver < V('
7.0.14'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

continued on next page

Hostname	IP Address	Score

Vulnerability

Vulnerability Name	MS-2024-Jul: .NET, .NET Framework, and Visual Studio Elevation of Privilege Vulnerability	**Score**	2
Published	2024-07-09	**Strategy**	Data-Driven Attack
	nCircle: 644518	**CVSS v2**	5.5
CVSS v3	7.3		

Description

DESCRIPTION
Microsoft .NET Framework and Visual Studios are subject to a elevation of privilege vulnerability. A local attacker could elevate privileges upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
Windows Registry

Advisory Publisher Entries

CVE:CVE-2024-38081	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-38081
CVSSv3 Base Score: 7.3	http://www.tripwire.com/vert/cvss/?data=7.3
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:
CWE: 59	http://cwe.mitre.org/data/definitions/59.html
MSRC Guidance: CVE-2024-38081	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-38081
Tripwire CVSSv3 Temporal Score: 7.1	http://www.tripwire.com/vert/cvss/?data=7.1
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)

continued on next page

Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1114 on 2024-07-10	http://www.tripwire.com/vert/?Released in ASPL 1114 on 2024-07-10

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9064" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9256.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302,11.0.2302.1" ) THEN C
ALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion
="2.0.50727.9177" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8977" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8977" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", file
Name="mscorlib.dll", startVersion="4.0.30319", patchedVersion="4.6.1947.0" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2
.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8977" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" )
THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersio
n="4.8.4739.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4101.0" )
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.32'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2021-May: Microsoft Jet Red Database Engine Remote Code Execution Vulnerability	**Score**	2
Published	2021-05-11	**Strategy**	Data-Driven Attack
	nCircle: 482848	**CVSS v2**	6.5
CVSS v3	8.8		

Description

DESCRIPTION
Microsoft Jet Red Database Engine is subject to a code execution vulnerability. A local attacker could execute code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

In addition to applying the patch, the registry key HKLM\SOFTWARE\Microsoft\Jet\4.0\Engines\AllowQueryRemoteTables must be configured to the value 0.

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

CVE:CVE-2021-28455	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-28455
CVSSv3 Base Score: 8.8	http://www.tripwire.com/vert/cvss/?data=8.8
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I
CWE: 77	http://cwe.mitre.org/data/definitions/77.html
MSRC Guidance: CVE-2021-28455	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2021-28455
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 943 on 2021-05-12	http://www.tripwire.com/vert/?Released in ASPL 943 on 2021-05-12

Rules

```
EXECUTE { import smb_file from version import Version as V, VersionException as VE
try:  win_ver = env.get
HostVariable( 'windows_version' ) hostis64 = env.getContextVariable( 'host_is_64_bit' ) except KeyError:
rule.STOP( False )
def get_file_version( path, file=r'system32\ntoskrnl.exe' ):  try:  path = r'
```

```
%s\\%s’ % (path,file) file_ver = smb_file.GetFileVersion(rule, None, path) ver = V(None, None,
file_ver) except (VE): rule.STOP(False) return ver
try:  path = env.getHostVariable(’wind
ows_system_root_directory’) except KeyError:  rule.STOP(False)
if win_ver.startswith( ’10.0.0.0’ ) and V(
’10.0’ ) <= get_file_version( path ) < V( ’10.0.10240.18932’ ):  rule.STOP(True) elif win_ver.startswith( ’
10.0.0.2’ ) and V( ’10.0’ ) <= get_file_version( path ) < V( ’10.0.14393.4402’ ):  rule.STOP(True) elif wi
n_ver.startswith( ’10.0.0.5’ ) and V( ’10.0’ ) <= get_file_version( path ) < V( ’10.0.17134.2208’ ):  rule
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2024-Apr: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability	**Score**	1
Published	2024-04-09	**Strategy**	Data-Driven Attack
	nCircle: 613962	**CVSS v2**	2.4
CVSS v3	7.3		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2024-21409	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-21409
CVSSv3 Base Score: 7.3	http://www.tripwire.com/vert/cvss/?data=7.3
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:H/I:
CWE: 416	http://cwe.mitre.org/data/definitions/416.html
MSRC Guidance: CVE-2024-21409	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-21409
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1101 on 2024-04-10	http://www.tripwire.com/vert/?Released in ASPL 1101 on 2024-04-10

continued on next page

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.dll
", startVersion="2.0.50727", patchedVersion="2.0.50727.9063" )
CALL isOSFamily( osFamily="10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="s
ystem.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8976" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="system.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9176" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion="4.8.9236.0" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="4.7", fileName="system.dll", startVersion="4.7", patchedVersion="4.7.4092.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" )
THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.dll", startVersion="4.8", patchedVersion=
"4.8.4718.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable(’PowerShell_Core_Version’) ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V(’7.0’) <= ver < V(’7.2.19’):  rule.STOP(True) elif V(’7.3’) <= ver < V(’7.3.12’):  rule.STOP(Tr
ue) elif V(’7.4’) <= ver < V(’7.4.2’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.29’):  rule.STOP(True) elif V(’7.0’) <= ver < V(’
7.0.18’):  rule.STOP(True) elif V(’8.0’) <= ver < V(’8.0.4’):  rule.STOP(True)
rule.STOP(Fa
lse) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2022-Dec: .NET Framework Remote Code Execution Vulnerability	**Score**	1
Published	2022-12-13	**Strategy**	Data-Driven Attack
	nCircle: 546521	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft PowerShell (via Microsoft Store)
Microsoft PowerShell (via SSH)
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
Windows Registry

Advisory Publisher Entries

CVE:CVE-2022-41089	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-41089
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
MSRC Guidance: CVE-2022-41089	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2022-41089
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1034 on 2022-12-14	http://www.tripwire.com/vert/?Released in ASPL 1034 on 2022-12-14

continued on next page

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.6", fileName="WPF\present
ationframework.dll", startVersion="4.6", patchedVersion="4.6.1888.0")
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="3.0", fileName="presentatio
nframework.dll", startVersion="3.0", patchedVersion="3.0.6920.9054")
CALL isOSFamily( osFamily="10.0.2009,10.0.2101,10.0.2102,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNetVu
lnerable(dotNetVersion="3.0", fileName="presentationframework.dll", startVersion="3.0", patchedVersion="3.0.69
20.9155")
CALL isOSFamily( osFamily="10.0.2009,10.0.2101,10.0.2102,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNetVu
lnerable(dotNetVersion="4.8", fileName="WPF\presentationframework.dll", startVersion="4.8", patchedVersion="4.
8.9115.0")
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="3.0", fileN
ame="presentationframework.dll", startVersion="3.0", patchedVersion="3.0.6920.8953")
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.7", fileN
ame="WPF\presentationframework.dll", startVersion="4.7", patchedVersion="4.7.4010.0")
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2101,10.0.2102,10.0.2202,11.0
.2102" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.8", fileName="WPF\presentationframework.dll", startVers
ion="4.8", patchedVersion="4.8.4590.0")
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  app_info = [(V(No
ne, None, ver), path) for ver, path in aspl_env.getContextVariable(’ms_store_pwsh_version’)] except (KeyError,
VE): rule.STOP(False)
for ver, path in app_info:  if V(’7.2’) <= ver < V(’7.2.9’):  rule.STOP(
True) elif V(’7.2’) <= ver < V(’7.3.2’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable(’PowerShell_Core_Version’) ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V(’7.2’) <= ver < V(’7.2.9’):  rule.STOP(True) elif V(’7.2’) <= ver < V(’7.3.2’):  rule.STOP(True
)
rule.STOP(False) }
EXECUTE { from version import Version, VersionException import aspl_env
try:  version = aspl_env.getContex
tVariable(’powershell_ssh_version’) ver = Version(version) except (KeyError, VersionException):  rule.S
TOP(False)
if Version(’7.2’) <= ver < Version(’7.2.9’):  rule.STOP(True) elif Version(’7.2’) <= ver < Vers
ion(’7.3.2’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’3.0’) <= ver < V(’3.1.32’):  rule.STOP(True) elif V(’6.0’) <= ver < V(’
6.0.12’):  rule.STOP(True) elif V(’7.0’) <= ver < V(’7.0.1’):  rule.STOP(True)
rule.STOP(Fa
lse) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
```

```
mKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Registry
...
```

```
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Jun: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability I	**Score**	1
Published	2023-06-13	**Strategy**	Data-Driven Attack
	nCircle: 581611	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2013
Microsoft Visual Studio 2015
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-24897	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-24897
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
CWE: 122	http://cwe.mitre.org/data/definitions/122.html
MSRC Guidance: CVE-2023-24897	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-24897
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9

continued on next page

Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1060 on 2023-06-14	http://www.tripwire.com/vert/?Released in ASPL 1060 on 2023-06-14

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="mscorlib.d
ll", startVersion="4.0.30319", patchedVersion="4.6.1912.0" )
CALL isOSFamily( osFamily="10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fi
leName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9058" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9166.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9171" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4050.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11
.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", pat
chedVersion="4.8.4644.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable(’PowerShell_Core_Version’) ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V(’7.2’) <= ver < V(’7.2.12’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.18’):  rule.STOP(True) elif V(’7.0’) <= ver < V(’
7.0.7’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
```

```
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file) else:
path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r'H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\14.0' rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=r'common7\ide\msdia140.dll')
if V('14.0') < ver < V('14
.0.24247.3'):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file) else:
path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r'H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\15.0' rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file='common7\\ide\\devenv.exe')
if V('15.0') < ver < V('15.
9.33801.237'):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file) else:
path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path =
r'HKLM\SOFTWARE\\Microsoft\VisualStudio\12.0\InstallDir' rule.RegistryGetValue(path) if not rule.success:
rule.STOP(False)
ver = get_file_version(rule.buffer, file=r'msdia120.dll')
if V('12.0') < ver < V('12.0.40
700.0'):  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Jun: .NET Framework Remote Code Execution Vulnerability	**Score**	1
Published	2023-06-13	**Strategy**	Data-Driven Attack
	nCircle: 581594	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-29326	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-29326
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
MSRC Guidance: CVE-2023-29326	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-29326
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1060 on 2023-06-14	http://www.tripwire.com/vert/?Released in ASPL 1060 on 2023-06-14

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="mscorlib.d
ll", startVersion="4.0.30319", patchedVersion="4.6.1912.0" )
CALL isOSFamily( osFamily="10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fi
leName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4050.0" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9058" )
CALL isOSFamily( osFamily="10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" ) THEN CALL isDotNetV
ulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.4644.0" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9166.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9171" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Jun: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability II	**Score**	1
Published	2023-06-13	**Strategy**	Data-Driven Attack
	nCircle: 581612	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft .NET Framework and Visual Studio are subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.0
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-24895	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-24895
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
MSRC Guidance: CVE-2023-24895	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-24895
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)

continued on next page

Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1060 on 2023-06-14	http://www.tripwire.com/vert/?Released in ASPL 1060 on 2023-06-14

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="mscorlib.d
ll", startVersion="4.0.30319", patchedVersion="4.6.1912.0" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9058" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9166.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9171" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4050.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2
102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patched
Version="4.8.4644.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable('PowerShell_Core_Version') ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V('7.2') <= ver < V('7.2.12'):  rule.STOP(True) if V('7.3') <= ver < V('7.3.5'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.18'):  rule.STOP(True) elif V('7.0') <= ver < V('
7.0.7'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability I	**Score**	1
Published	2023-09-12	**Strategy**	Data-Driven Attack
	nCircle: 588865	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft Visual Studios and .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2013
Microsoft Visual Studio 2015
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36792	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36792
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
CWE: 190	http://cwe.mitre.org/data/definitions/190.html
MSRC Guidance: CVE-2023-36792	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36792
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9

continued on next page

Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1073 on 2023-09-13	http://www.tripwire.com/vert/?Released in ASPL 1073 on 2023-09-13

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9061" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9174" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8",
fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4662.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVers
ion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9181.0" )
CALL isOSFamily( osFamily="10.0.2102.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.
web.dll", startVersion="4.8", patchedVersion="4.8.9186.0" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.
web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102.1" ) THEN CALL isDotNetVulnerable(
dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4667.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fileName
="system.web.dll", startVersion="4.7", patchedVersion="4.7.4063.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable(’PowerShell_Core_Version’) ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V(’7.2’) <= ver < V(’7.2.12’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.22’):  rule.STOP(True) elif V(’7.0’) <= ver < V(’
7.0.11’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file) else:
path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r'H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\14.0' rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=r'common7\ide\msdia140.dll')
if V('14.0') < ver < V('14
.0.24248.0'):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file) else:
path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r'H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\15.0' rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file='common7\\ide\\devenv.exe')
if V('15.0') < ver < V('15.
9.34031.82'):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file) else:
path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path =
r'HKLM\SOFTWARE\\Microsoft\VisualStudio\12.0\InstallDir' rule.RegistryGetValue(path) if not rule.success:
rule.STOP(False)
ver = get_file_version(rule.buffer, file=r'msdia120.dll')
if V('12.0') < ver < V('12.0.40
707.5'):  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability II	**Score**	1
Published	2023-09-12	**Strategy**	Data-Driven Attack
	nCircle: 588880	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft Visual Studios and .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2013
Microsoft Visual Studio 2015
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36796	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36796
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
CWE: 191	http://cwe.mitre.org/data/definitions/191.html
MSRC Guidance: CVE-2023-36796	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36796
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9

continued on next page

Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1073 on 2023-09-13	http://www.tripwire.com/vert/?Released in ASPL 1073 on 2023-09-13

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9061" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9174" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8",
fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4662.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVers
ion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9181.0" )
CALL isOSFamily( osFamily="10.0.2102.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.
web.dll", startVersion="4.8", patchedVersion="4.8.9186.0" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.
web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102.1" ) THEN CALL isDotNetVulnerable(
dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4667.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fileName
="system.web.dll", startVersion="4.7", patchedVersion="4.7.4063.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable('PowerShell_Core_Version') ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V('7.2') <= ver < V('7.2.12'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.22'):  rule.STOP(True) elif V('7.0') <= ver < V('
7.0.11'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\14.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=r’common7\ide\msdia140.dll’)
if V(’14.0’) < ver < V(’14
.0.24248.0’):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\15.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=’common7\\ide\\devenv.exe’)
if V(’15.0’) < ver < V(’15.
9.34031.82’):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path =
r’HKLM\SOFTWARE\\Microsoft\VisualStudio\12.0\InstallDir’ rule.RegistryGetValue(path) if not rule.success:
rule.STOP(False)
ver = get_file_version(rule.buffer, file=r’msdia120.dll’)
if V(’12.0’) < ver < V(’12.0.40
707.5’):  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability III	**Score**	1
Published	2023-09-12	**Strategy**	Data-Driven Attack
	nCircle: 588881	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft Visual Studios and .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2013
Microsoft Visual Studio 2015
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36794	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36794
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
CWE: 191	http://cwe.mitre.org/data/definitions/191.html
MSRC Guidance: CVE-2023-36794	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36794
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9

continued on next page

Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1073 on 2023-09-13	http://www.tripwire.com/vert/?Released in ASPL 1073 on 2023-09-13

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9061" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9174" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8",
fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4662.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVers
ion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9181.0" )
CALL isOSFamily( osFamily="10.0.2102.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.
web.dll", startVersion="4.8", patchedVersion="4.8.9186.0" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.
web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102.1" ) THEN CALL isDotNetVulnerable(
dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4667.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fileName
="system.web.dll", startVersion="4.7", patchedVersion="4.7.4063.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable(’PowerShell_Core_Version’) ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V(’7.2’) <= ver < V(’7.2.12’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.22’):  rule.STOP(True) elif V(’7.0’) <= ver < V(’
7.0.11’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\14.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=r’common7\ide\msdia140.dll’)
if V(’14.0’) < ver < V(’14
.0.24248.0’):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\15.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=’common7\\ide\\devenv.exe’)
if V(’15.0’) < ver < V(’15.
9.34031.82’):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path =
r’HKLM\SOFTWARE\\Microsoft\VisualStudio\12.0\InstallDir’ rule.RegistryGetValue(path) if not rule.success:
rule.STOP(False)
ver = get_file_version(rule.buffer, file=r’msdia120.dll’)
if V(’12.0’) < ver < V(’12.0.40
707.5’):  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Sep: Visual Studio Remote Code Execution Vulnerability IV	**Score**	1
Published	2023-09-12	**Strategy**	Data-Driven Attack
	nCircle: 588884	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft Visual Studios and .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2013
Microsoft Visual Studio 2015
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36793	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36793
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
CWE: 122	http://cwe.mitre.org/data/definitions/122.html
MSRC Guidance: CVE-2023-36793	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36793
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9

continued on next page

Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1073 on 2023-09-13	http://www.tripwire.com/vert/?Released in ASPL 1073 on 2023-09-13

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9061" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9174" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8",
fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4662.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVers
ion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9181.0" )
CALL isOSFamily( osFamily="10.0.2102.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.
web.dll", startVersion="4.8", patchedVersion="4.8.9186.0" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.
web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fileName
="system.web.dll", startVersion="4.7", patchedVersion="4.7.4063.0" )
CALL isOSFamily( osFamily="6.1,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102.1" ) THEN CALL isDotNetVulnerable( dot
NetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4667.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable('PowerShell_Core_Version') ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V('7.2') <= ver < V('7.2.12'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.22'):  rule.STOP(True) elif V('7.0') <= ver < V('
7.0.11'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\14.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=r’common7\ide\msdia140.dll’)
if V(’14.0’) < ver < V(’14
.0.24248.0’):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\15.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=’common7\\ide\\devenv.exe’)
if V(’15.0’) < ver < V(’15.
9.34031.82’):  rule.STOP(True)
rule.STOP(False) }
```

```
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path =
r’HKLM\SOFTWARE\\Microsoft\VisualStudio\12.0\InstallDir’ rule.RegistryGetValue(path) if not rule.success:
rule.STOP(False)
ver = get_file_version(rule.buffer, file=r’msdia120.dll’)
if V(’12.0’) < ver < V(’12.0.40
707.5’):  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Sep: .NET Framework Remote Code Execution Vulnerability	**Score**	1
Published	2023-09-12	**Strategy**	Data-Driven Attack
	nCircle: 588915	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36788	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36788
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:L	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
MSRC Guidance: CVE-2023-36788	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36788
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1073 on 2023-09-13	http://www.tripwire.com/vert/?Released in ASPL 1073 on 2023-09-13

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9061" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fileName="system.web.dll", startVersion="4.7", patchedVersion="4.7.4063.0" )
CALL isOSFamily( osFamily="10.0.0.6,10.0.2102.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4667.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9174" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4662.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9181.0" )
CALL isOSFamily( osFamily="10.0.2102.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9186.0" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
CALL isOSFamily( osFamily="6.1,6.2,6.3" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8974" )
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Nov: ASP.NET Security Feature Bypass Vulnerability	**Score**	1
Published	2023-11-14	**Strategy**	Data-Driven Attack
	nCircle: 594060	**CVSS v2**	2.4
CVSS v3	8.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a security feature bypass vulnerability. A local attacker could bypass security checks upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36560	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36560
CVSSv3 Base Score: 8.8	http://www.tripwire.com/vert/cvss/?data=8.8
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I
MSRC Guidance: CVE-2023-36560	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36560
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1082 on 2023-11-15	http://www.tripwire.com/vert/?Released in ASPL 1082 on 2023-11-15

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="system.web
.dll", startVersion="4.0.30319", patchedVersion="4.6.1929.0" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9062" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302
.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patche
dVersion="4.8.9206.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302.0,11.0.2302.1" ) THEN
CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVer
sion="2.0.50727.9175" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8975" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll"
, startVersion="2.0.50727", patchedVersion="2.0.50727.8975" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8975" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11
.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", p
atchedVersion="4.8.4682.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="4.7", fileName="system.web.dll", startVersion="4.7", patchedVersion="4.7.4076.0" )
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Feb: .NET and Visual Studio Remote Code Execution Vulnerability	**Score**	1
Published	2023-02-14	**Strategy**	Data-Driven Attack
	nCircle: 554440	**CVSS v2**	2.4
CVSS v3	7.8		

Description

DESCRIPTION
.NET Framework and Visual Studio is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2013
Microsoft Visual Studio 2015
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-21808	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-21808
CVSSv3 Base Score: 7.8	http://www.tripwire.com/vert/cvss/?data=7.8
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:
CWE: 416	http://cwe.mitre.org/data/definitions/416.html
MSRC Guidance: CVE-2023-21808	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-21808
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9

continued on next page

Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1042 on 2023-02-15	http://www.tripwire.com/vert/?Released in ASPL 1042 on 2023-02-15

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.6", fileName="mscorlib.dl
l", startVersion="4.0.30319", patchedVersion="4.6.1901.0")
CALL isOSFamily( osFamily="10.0.0.0,10.0.0.2,10.0.1.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="2.0", fil
eName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8966")
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="2.0", fileName="mscorlib.dl
l", startVersion="2.0.50727", patchedVersion="2.0.50727.9055")
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2102.1,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable(dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9
168")
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable(dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9139.0")
CALL isOSFamily( osFamily="6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="4
.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4038.0")
CALL isOSFamily( osFamily="6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2
102" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedV
ersion="4.8.4614.0")
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable('PowerShell_Core_Version') ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V('7.0') <= ver < V('7.2.10'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.14'):  rule.STOP(True) elif V('7.0') <= ver < V('
7.0.3'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
```

```
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\14.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=r’common7\ide\msdia140.dll’)
if V(’14.0’) < ver < V(’14
.0.24247.3’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r’H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\15.0’ rule.RegistryGetValue(path) if not rule.success:  rule.S
TOP(False)
ver = get_file_version(rule.buffer, file=’common7\\ide\\devenv.exe’)
if V(’15.0’) < ver < V(’15.
9.33403.129’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file) else:
path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path =
r’HKLM\SOFTWARE\\Microsoft\VisualStudio\12.0\InstallDir’ rule.RegistryGetValue(path) if not rule.success:
rule.STOP(False)
ver = get_file_version(rule.buffer, file=r’msdia120.dll’)
if V(’12.0’) < ver < V(’12.0.40
700.0’):  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	No UNC Paths Configured for Privacy	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 205863	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
There are no hardened UNC paths configured in Group Policy to require the use of RequirePrivacy.
SOLUTION
Configure hardened UNC paths in Group Policy to use the RequirePrivacy flag as seen in http://support.microsoft.com/kb/3000483.

Affected Applications

Application Name
Windows Domain Joined Host

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 601 on 2015-02-11	http://www.tripwire.com/vert/?Released in ASPL 601 on 2015-02-11

Rules

```
EXECUTE { try:  hardened = env.getHostVariable(’hardened_unc_paths’) if len(hardened) == 0:  rul
e.STOP(True) except KeyError:  rule.STOP(False)
match = True if hardened:  for unc in hardened:
if hardened[unc][’privacy’] == 1:  match = False
rule.STOP(match) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	No UNC Paths Configured for Mutual Authentication	**Score**	0
Published	nCircle: 205864	**Strategy**	Data-Driven Attack
CVSS v3	0.0	**CVSS v2**	0.0

Description

DESCRIPTION
There are no hardened UNC paths configured in Group Policy to require the use of Mutual Authentication.
SOLUTION
Configure hardened UNC paths in Group Policy to use the RequireAuthentication flag as seen in http://support.microsoft.com/kb/3000483.

Affected Applications

Application Name
Windows Domain Joined Host

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 601 on 2015-02-11	http://www.tripwire.com/vert/?Released in ASPL 601 on 2015-02-11

Rules

```
EXECUTE { try:  hardened = env.getHostVariable('hardened_unc_paths') if len(hardened) == 0:  rul
e.STOP(True) except KeyError:  rule.STOP(False)
match = True if hardened:  for unc in hardened:
if hardened[unc]['authentication'] == 1:  match = False
rule.STOP(match) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Windows DRT Command Success	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 211953	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
IP360 was able to successfully access the registry and/or file system using the provided credentials.

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:U/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:U/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 615 on 2015-05-16	http://www.tripwire.com/vert/?Released in ASPL 615 on 2015-05-16

Rules

```
EXECUTE{ import smb_file, HIC registry_access = False cifs_system_access = False rule.RegistryGetValue(r'HKLM\
Software\Microsoft\Windows NT\CurrentVersion\SystemRoot')
data = 'Windows Registry Access:  %s, CIFS Filesyste
m Access:  %s'
if rule.success:  registry_access = True smb_file.CheckPathExists(rule, '', rule.buf
fer) if rule.success:  cifs_system_access = True
data = data % ( str( registry_access ), str( cifs
_system_access ) ) HIC.insert_host_data(env.target, 'windows_drt_access', 'WDRT', data) if cifs_system_access
and registry_access:  rule.STOP( True ) rule.STOP( False ) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	RFC7525 Recommended Cipher Suites Exposure	**Score**	0
Published	nCircle: 213235	**Strategy**	Access Control Breach
CVSS v3	0.0	**CVSS v2**	0.0

Description

DESCRIPTION
RFC7525 "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)" section 4.2 states that servers should implement and deploy TLS_DHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_DHE_RSA_WITH_AES_256_GCM_SHA384, and TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.
SOLUTION
Refer to vendor documentation for instructions on how to configure ciphersuite preferences.

Affected Applications

Application Name
TLSv1.2

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 623 on 2015-07-09	http://www.tripwire.com/vert/?Released in ASPL 623 on 2015-07-09

Rules

```
EXECUTE { import aspl_env try:  lstCiphers = aspl_env.getContextVariable("tlsv1.2_accepted_ciphers") except
KeyError:  rule.STOP(False)
rfc7525_cipher = dict()
rfc7525_cipher['\x00\x9E'] = "TLS_DHE_RSA_WITH_AES_1
28_GCM_SHA256" rfc7525_cipher['\xC0\x2F'] = "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256" rfc7525_cipher['\x00\x9F']
= "TLS_DHE_RSA_WITH_AES_256_GCM_SHA384" rfc7525_cipher['\xC0\x30'] = "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"
for cipher in rfc7525_cipher.keys():  if cipher not in lstCiphers:  rule.STOP(True)
rule.STOP(Fals
e) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS15-124: Microsoft Browser ASLR Bypass Vulnerability	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 220130	**CVSS v2**	4.3
CVSS v3	0.0		

Description

DESCRIPTION
Microsoft Browser contains an ASLR Bypass Vulnerability. The vulnerability could allow an attacker to bypass the Address Space Layout Randomization (ASLR) security feature.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft Internet Explorer 10
Microsoft Internet Explorer 11
Microsoft Internet Explorer 7
Microsoft Internet Explorer 8
Microsoft Internet Explorer 9
Windows Registry

Advisory Publisher Entries

BugTraq: 78537	http://www.securityfocus.com/bid/78537
CVE:CVE-2015-6161	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-6161
CWE: 200	http://cwe.mitre.org/data/definitions/200.html
MS Advisory Number: MS15-124	http://technet.microsoft.com/en-us/security/bulletin/MS15-124
MS Hotfix Number: 3104002	http://support.microsoft.com/default.aspx?scid=KB;en-us;3104002
Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 648 on 2015-12-09	http://www.tripwire.com/vert/?Released in ASPL 648 on 2015-12-09

Rules

```
EXECUTE { from smbutil import getKnownFileVersionObj from version import Version as V, VersionException import
smb_file import aspl_env
def get_file_version(system_root, file = ’win32k.sys’):  try:  path = ’%s
\\system32\\%s’ % (system_root,file) file_ver = smb_file.GetFileVersion(rule, None, path) ver
= V(None, None, file_ver) except VersionException:  rule.STOP(False) return ver
try:
```

```
win_ver = aspl_env.getHostVariable(’windows_version’) system_root = env.getHostVariable(’windows_system
_root_directory’) except KeyError:  rule.STOP( False )
try:  is64 = env.getContextVariable(’host_is_64_
bit’) except KeyError:  is64 = False
keys = [r’HKLM\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureContr
ol\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iexplore.exe’] if is64:  keys.append(r’HKLM\SOFTWARE\Wo
w6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iex
...
```

```
EXECUTE { from smbutil import getKnownFileVersionObj from version import Version as V, VersionException import
smb_file import aspl_env
def get_file_version(system_root, file = ’win32k.sys’):  try:  path = ’%s
\\system32\\%s’ % (system_root,file) file_ver = smb_file.GetFileVersion(rule, None, path) ver
= V(None, None, file_ver) except VersionException:  rule.STOP(False) return ver
try:
win_ver = aspl_env.getHostVariable(’windows_version’) system_root = env.getHostVariable(’windows_system
_root_directory’) except KeyError:  rule.STOP( False )
try:  is64 = env.getContextVariable(’host_is_64_
bit’) except KeyError:  is64 = False
keys = [r’HKLM\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureContr
ol\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iexplore.exe’] if is64:  keys.append(r’HKLM\SOFTWARE\Wo
w6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iex
...
```

```
EXECUTE { from smbutil import getKnownFileVersionObj from version import Version as V, VersionException import
smb_file import aspl_env
def get_file_version(system_root, file = ’win32k.sys’):  try:  path = ’%s
\\system32\\%s’ % (system_root,file) file_ver = smb_file.GetFileVersion(rule, None, path) ver
= V(None, None, file_ver) except VersionException:  rule.STOP(False) return ver
try:
win_ver = aspl_env.getHostVariable(’windows_version’) system_root = env.getHostVariable(’windows_system
_root_directory’) except KeyError:  rule.STOP( False )
try:  is64 = env.getContextVariable(’host_is_64_
bit’) except KeyError:  is64 = False
keys = [r’HKLM\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureContr
ol\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iexplore.exe’] if is64:  keys.append(r’HKLM\SOFTWARE\Wo
w6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iex
...
```

```
EXECUTE { from smbutil import getKnownFileVersionObj from version import Version as V, VersionException import
smb_file import aspl_env
def get_file_version(system_root, file = ’win32k.sys’):  try:  path = ’%s
\\system32\\%s’ % (system_root,file) file_ver = smb_file.GetFileVersion(rule, None, path) ver
= V(None, None, file_ver) except VersionException:  rule.STOP(False) return ver
try:
win_ver = aspl_env.getHostVariable(’windows_version’) system_root = env.getHostVariable(’windows_system
_root_directory’) except KeyError:  rule.STOP( False )
try:  is64 = env.getContextVariable(’host_is_64_
bit’) except KeyError:  is64 = False
keys = [r’HKLM\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureContr
ol\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iexplore.exe’] if is64:  keys.append(r’HKLM\SOFTWARE\Wo
w6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iex
...
```

```
EXECUTE { from smbutil import getKnownFileVersionObj from version import Version as V, VersionException import
smb_file import aspl_env
def get_file_version(system_root, file = ’win32k.sys’):  try:  path = ’%s
\\system32\\%s’ % (system_root,file) file_ver = smb_file.GetFileVersion(rule, None, path) ver
= V(None, None, file_ver) except VersionException:  rule.STOP(False) return ver
try:
win_ver = aspl_env.getHostVariable(’windows_version’) system_root = env.getHostVariable(’windows_system
_root_directory’) except KeyError:  rule.STOP( False )
try:  is64 = env.getContextVariable(’host_is_64_
bit’) except KeyError:  is64 = False
keys = [r’HKLM\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureContr
ol\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iexplore.exe’] if is64:  keys.append(r’HKLM\SOFTWARE\Wo
w6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_ALLOW_USER32_EXCEPTION_HANDLER_HARDENING\iex
```

```
. . .
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Perfect Forward Secrecy Preferred	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 279476	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The service implements a perfect forward secrecy enabled ciphersuite as the preferred cipher. Previously intercepted TLS connections using this cipher forward secrecy cannot be decrypted by an attacker with access to long-term keys. The use of ephemeral keys allows for forward secrecy by protecting each session with unique key material such that an attacker would have to crack the encryption for each session individually.

This condition is tested by sending a TLS Client Hello featuring support similar to Chrome 80 on Windows 10 and observing which cipher is selected by the server.
SOLUTION
This is an informational check indicating an ideal configuration. No change is suggested.

Affected Applications

Application Name
SSL

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 693 on 2016-10-12	http://www.tripwire.com/vert/?Released in ASPL 693 on 2016-10-12

Rules

```
EXECUTE { import aspl_env from aspl_tls_enumerator import getHostnames, TLSV13 match_hosts = [] try:  refer
ence_hello = env.getContextVariable('ReferenceBrowserServerHello') except KeyError:  rule.STOP(False)
for
hostname in reference_hello:  if 'DHE' in reference_hello[hostname][2] or reference_hello[hostname][0] == T
LSV13:  match_hosts += [hostname]
if len(match_hosts) > 0:  rule.appendTranscript("The following ho
stnames prefer PFS: %s" % (match_hosts)) rule.STOP(match_hosts) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Perfect Forward Secrecy Available	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 279477	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The service implements a perfect forward secrecy enabled ciphersuite. Previously intercepted TLS connections using a forward secrecy cipher cannot be decrypted by an attacker with access to long-term keys. The use of ephemeral keys allows for forward secrecy by protecting each session with unique key material such that an attacker would have to crack the encryption for each session individually.
SOLUTION
This is an informational check. No change is suggested.

Affected Applications

Application Name
SSL

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 693 on 2016-10-12	http://www.tripwire.com/vert/?Released in ASPL 693 on 2016-10-12

Rules

```
EXECUTE { from aspl_env import getContextVariable from aspl_tls13 import TLSV13 from aspl_ssl import ssl3_ciph
er
try:  protocols = getContextVariable(’supported_ciphers_by_protocol’) except KeyError:  rule.STOP(Fa
lse)
try:  if len(protocols[TLSV13]) > 0:  rule.STOP(True) except KeyError:  pass for protocol i
n protocols:  for cipher in protocols[protocol]:  if cipher in ssl3_cipher and ’DHE’ in ssl3_
cipher[cipher]:  rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Google Chrome Enterprise Policy Site Isolation Per Process Not Enabled	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 316523	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
Google Chrome's Site Isolation policy has not been enabled on a per process basis via the Enterprise Policy. Site isolation helps prevent sensitive data from being leaked to a malicious site attempting to bypass the same-origin policy.

NOTE: Alternative mitigation options including enabling Site Isolation per site via the Enterprise Policy, and per user configuration via the Google Chrome app is not supported by this check.
SOLUTION
The vendor has provided steps to enable Site Isolation per process in Google Chrome version 63.x and later. Please see the provided link for more information.

Affected Applications

Application Name
Google Chrome

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 759 on 2018-01-05	http://www.tripwire.com/vert/?Released in ASPL 759 on 2018-01-05

Rules

```
EXECUTE { import aspl_env from version import Version as V
try:  ver = aspl_env.getContextVariable('chrome
Version')
if V(ver) < V('63.0'):  rule.STOP(True)
except KeyError:  pass
rule.Registr
yGetValue('HKLM\Software\Policies\Google\Chrome\SitePerProcess')
rule.STOP( not (rule.buffer and rule.buffer
== '0x00000001') )
}
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	TLSv1.2 Enabled	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 419410	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
TLSv1.2 is enabled on this host.
SOLUTION
This is an informational check only.

Affected Applications

Application Name
TLSv1.2

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 822 on 2019-03-19	http://www.tripwire.com/vert/?Released in ASPL 822 on 2019-03-19

Rules

```
STOP WITH Match
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Remote Desktop Network Level Authentication (NLA) Enabled	**Score**	0
Published	nCircle: 423483	**Strategy**	Network Reconnaissance
CVSS v3	0.0	**CVSS v2**	0.0

Description

DESCRIPTION
Network Level Authentication is enabled. This is a recommended mitigation.
SOLUTION
No action is required.

Affected Applications

Application Name
Microsoft Remote Desktop Protocol

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:U/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:U/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 832 on 2019-05-29	http://www.tripwire.com/vert/?Released in ASPL 832 on 2019-05-29

Rules

```
EXECUTE { try:  rdp_proto = env.getContextVariable('rdp_protocol_version') if rdp_proto == '\x05\x00\x0
0\x00':  rule.STOP(True) except KeyError:  pass rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	CACHED APPLICATION DATA	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 479266	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The instance data of this vulnerability contains the data stored in the cache after the application scan.

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:U/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:U/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 937 on 2021-03-30	http://www.tripwire.com/vert/?Released in ASPL 937 on 2021-03-30

Rules

```
EXECUTE { try:  data = env.getContextVariable(’ASPLCache’)[0] pretty_data = ’’ try:  for que
ry, item in data:  pretty_data += ’%s %s\n’ % (query, item) pretty_data += ’\t%s\n’ % s
tr(data[(query, item)]) except MemoryError:  pass rule.transcript = pretty_data rule.transc
riptIsFull = True except KeyError:  pass }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	DCE RPC mapper available	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 1225	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
DCE is Microsoft's implementation of the RPC protocol.

Microsoft uses DCE in the same manner that Unix uses portmap. This service is used to register other services with a central control program that facilitates distributed computing.

This service can be used by an attacker to determine the name, version, and location of any DCOM or RPC service on the machine.

Affected Applications

Application Name
DCE/MS RPC over TCP

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
STOP WITH Match
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MIME Type Sniffing Disabled	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 507122	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The remote server specifies the nosniff X-Content-Type-Option on one or more vhosts.
SOLUTION
This is an informational check. No configuration change is needed.

Affected Applications

Application Name
HTTP

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:U/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:U/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 961 on 2021-08-24	http://www.tripwire.com/vert/?Released in ASPL 961 on 2021-08-24

Rules

```
EXECUTE { from aspl_env import getContextVariable
thisHeader = 'X_ContentType' expectedValueSubstring = 'nosn
iff'
try:  Headers = getContextVariable('HTTP_Headers') except KeyError:  rule.STOP(False)
TranscriptM
essage = 'MIME type sniffing is disabled for the following vhosts:  '  EnabledHosts = [] for hostname in Headers
:  if thisHeader in Headers[hostname]:  if not expectedValueSubstring in Headers[hostname][thisHeade
r]:  continue if hostname is None:  EnabledHosts += ['(default)'] else:
EnabledHosts += [hostname]
if len(EnabledHosts) > 0:  rule.transcript = TranscriptMessage + ',
'.join(EnabledHosts) rule.STOP(True) rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

continued on next page

Hostname	IP Address	Score

Vulnerability

Vulnerability Name	ms-msdt Protocol Scheme Configured	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 529971	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The ms-msdt protocol scheme is configured on this system. This protocol scheme has been associated with the Follina vulnerability allowing for remote code execution within Microsoft Office.
SOLUTION
Protocol Schemes can be deleted from the registry (HKCR) to remove the association.

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1005 on 2022-05-31	http://www.tripwire.com/vert/?Released in ASPL 1005 on 2022-05-31

Rules

```
RegistryQuery GetKey[HKCR\ms-msdt] THEN CHECK Exists
```

Hosts

Hostname	**IP Address**	**Score**
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	search-ms Protocol Scheme Configured	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 530236	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The search-ms protocol scheme is configured on this system. This protocol scheme can allow an attacker to open an Explorer window which points at a remote share with a custom display name, potentially allowing the end user to be social engineered.
SOLUTION
Protocol Schemes can be deleted from the registry (HKCR) to remove the association.

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1006 on 2022-06-04	http://www.tripwire.com/vert/?Released in ASPL 1006 on 2022-06-04

Rules

```
RegistryQuery GetKey[HKCR\search-ms] THEN CHECK Exists
```

Hosts

Hostname	**IP Address**	**Score**
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Unquoted Service Path Weakness	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 530548	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
A vulnerability exists due to the way in which the CreateProcess function creates new processes. When a process path contains spaces, the CreateProcess function attempts to execute a process at each point where a spaces occurs. For example, in the path C:\Program Files\Tripwire Demo\example.exe, the CreateProcess function will attempt to execute C:\Program.exe and C:\Program Files\Tripwire.exe before trying C:\Program Files\Tripwire Demo\example.exe.

This vulnerability can be exploited when services do not properly enclose paths with spaces within quotes.
SOLUTION
Ensure that all executable service paths are wrapped in quotes.

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

CWE: 428	http://cwe.mitre.org/data/definitions/428.html
Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1007 on 2022-06-15	http://www.tripwire.com/vert/?Released in ASPL 1007 on 2022-06-15

Rules

```
EXECUTE { import HIC import aspl_env
reg_path = ’HKLM\\System\\CurrentControlSet\\Services’ try:  system_root
= aspl_env.getHostVariable(’windows_system_root_directory’).lower() except KeyError:  rule.STOP(False) unquote
d_paths = [] services = [] system_paths = [ ’%systemroot%\\system32\\svchost.exe ’, ’%systemroot%\\sys
tem32\\dllhost.exe ’, ’%systemroot%\\system32\\msiexec.exe ’, ]
def test_unquoted_path(path, modified
_path):  if ’ ’ not in path:  return False elif filter_system_paths(path):  return False
elif path.startswith(’"’) and path.endswith(’"’):  return False elif path.startswith("’") and p
ath.endswith("’"):  return False elif split_and_test(path, "’"):  return False elif spl
it_and_test(path, ’"’):  return False elif find_valid_spaces(modified_path):  return False
return True
def filter_system_paths(path):  for system_path in system_paths:  if path.startswith
```

```
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2022-Nov: .NET Framework Information Disclosure Vulnerability	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 542938	**CVSS v2**	0.0
CVSS v3	5.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to an information disclosure vulnerability. A local attacker could access queries from other users in the SQL Connection Pool upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2022-41064	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-41064
CVSSv3 Base Score: 5.8	http://www.tripwire.com/vert/cvss/?data=5.8
CVSSv3 Base Vector: CVSS:3.1/AV:A/AC:H/PR:L/UI:N/S:(	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:A/AC:H/PR:L/UI:N/S:C/C:H/I
MSRC Guidance: CVE-2022-41064	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2022-41064
Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1029 on 2022-11-09	http://www.tripwire.com/vert/?Released in ASPL 1029 on 2022-11-09

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.6", fileName="WPF\present
ationframework.dll", startVersion="4.6", patchedVersion="4.6.1810.0")
CALL isOSFamily( osFamily="10.0.2009,10.0.2101,10.0.2102,10.0.2202,11.0.2102" ) THEN CALL isDotNetVulnerable(d
otNetVersion="4.8", fileName="WPF\presentationframework.dll", startVersion="4.8", patchedVersion="4.8.4579.0")
```

```
CALL isOSFamily( osFamily="10.0.2009,10.0.2101,10.0.2102,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.8", fileName="WPF\presentationframework.dll", startVersion="4.8", patchedVersion="4.8.9105.0")
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.7", fileName="WPF\presentationframework.dll", startVersion="4.7", patchedVersion="4.7.4005.0")
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.8", fileName="WPF\presentationframework.dll", startVersion="4.8", patchedVersion="4.8.4585.0")
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Feb: .NET Framework Denial of Service Vulnerability	**Score**	0
Published	2023-02-14	**Strategy**	DoS
	nCircle: 554439	**CVSS v2**	2.1
CVSS v3	5.0		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a denial of service vulnerability. An attacker could cause a denial of service condition upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-21722	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-21722
CVSSv3 Base Score: 5	http://www.tripwire.com/vert/cvss/?data=5
CVSSv3 Base Vector: CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:L/AC:L/PR:L/UI:R/S:U/C:N/I:
CWE: 59	http://cwe.mitre.org/data/definitions/59.html
MSRC Guidance: CVE-2023-21722	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-21722
Tripwire CVSSv3 Temporal Score: 8.5	http://www.tripwire.com/vert/cvss/?data=8.5
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1042 on 2023-02-15	http://www.tripwire.com/vert/?Released in ASPL 1042 on 2023-02-15

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.6", fileName="mscorlib.dl
l", startVersion="4.0.30319", patchedVersion="4.6.1901.0")
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="2.0", fileName="mscorlib.dl
l", startVersion="2.0.50727", patchedVersion="2.0.50727.9055")
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2102.1,10.0.2202,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable(dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9
168")
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable(dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9139.0")
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="2.0", fileName="mscorlib.dll", s
tartVersion="2.0.50727", patchedVersion="2.0.50727.8966")
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="2.0", fileName="mscorlib.dll", s
tartVersion="2.0.50727", patchedVersion="2.0.50727.8966")
CALL isOSFamily( osFamily="6.2,6.3,10.0.0.0,10.0.0.2,10.0.1.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="2
.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8966")
CALL isOSFamily( osFamily="6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable(dotNetVersion="4
.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4038.0")
CALL isOSFamily( osFamily="6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2
102" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedV
ersion="4.8.4614.0")
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Jun: .NET and Visual Studio Denial of Service Vulnerability	**Score**	0
Published	2023-06-13	**Strategy**	DoS
	nCircle: 581558	**CVSS v2**	2.1
CVSS v3	7.5		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a denial of service vulnerability. An attacker could cause a denial of service condition upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-32030	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-32030
CVSSv3 Base Score: 7.5	http://www.tripwire.com/vert/cvss/?data=7.5
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I
MSRC Guidance: CVE-2023-32030	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-32030
Tripwire CVSSv3 Temporal Score: 5.2	http://www.tripwire.com/vert/cvss/?data=5.2
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1060 on 2023-06-14	http://www.tripwire.com/vert/?Released in ASPL 1060 on 2023-06-14

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="mscorlib.dll", startVersion="4.0.30319", patchedVersion="4.6.1912.0" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9058" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9166.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9171" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4050.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.4644.0" )
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Jun: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability	**Score**	0
Published	2023-06-13	**Strategy**	DoS
	nCircle: 581585	**CVSS v2**	2.1
CVSS v3	7.5		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a denial of service vulnerability. An attacker could cause a denial of service condition upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
PowerShell Core
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-29331	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-29331
CVSSv3 Base Score: 7.5	http://www.tripwire.com/vert/cvss/?data=7.5
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I
CWE: 400	http://cwe.mitre.org/data/definitions/400.html
MSRC Guidance: CVE-2023-29331	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-29331
Tripwire CVSSv3 Temporal Score: 5.2	http://www.tripwire.com/vert/cvss/?data=5.2
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)

continued on next page

Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1060 on 2023-06-14	http://www.tripwire.com/vert/?Released in ASPL 1060 on 2023-06-14

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.6", fileName="mscorlib.d
ll", startVersion="4.0.30319", patchedVersion="4.6.1912.0" )
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9058" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9166.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable
( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9171" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4050.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.0,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8970" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11
.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", pat
chedVersion="4.8.4644.0" )
EXECUTE { from version import Version as V, VersionException as VE import aspl_env
try:  version = aspl_en
v.getContextVariable('PowerShell_Core_Version') ver = V(version) except (KeyError, VE): rule.STOP(Fals
e)
if V('7.2') <= ver < V('7.2.12'):  rule.STOP(True) if V('7.3') <= ver < V('7.3.5'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('6.0') <= ver < V('6.0.18'):  rule.STOP(True) elif V('7.0') <= ver < V('
7.0.7'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2023-Aug: .NET Framework Spoofing Vulnerability	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 585544	**CVSS v2**	0.0
CVSS v3	5.9		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a spoofing vulnerability. For successful exploitation would require an attacker to create a crafted certificate in order to validate themselves as a trusted source.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Windows Registry

Advisory Publisher Entries

CVE:CVE-2023-36873	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-36873
CVSSv3 Base Score: 5.9	http://www.tripwire.com/vert/cvss/?data=5.9
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/
CWE: 20	http://cwe.mitre.org/data/definitions/20.html
MSRC Guidance: CVE-2023-36873	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2023-36873
Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1068 on 2023-08-09	http://www.tripwire.com/vert/?Released in ASPL 1068 on 2023-08-09

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web
.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9061" )
CALL isOSFamily( osFamily="10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.9176.0" )
```

```
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="system.web.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.9174" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.7", fileName="system.web.dll", startVersion="4.7", patchedVersion="4.7.4057.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2009,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="system.web.dll", startVersion="4.8", patchedVersion="4.8.4654.0" )
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	SSL Server Supports CBC Ciphers for TLSv1.2	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 602422	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
Cipher Block Chaining (CBC) is vulnerable to beast attacks. BEAST attack relies on a weakness in the way CBC mode is used in SSL and TLS.
SOLUTION
Disable any Cipher Suites using CBC ciphers.

Affected Applications

Application Name
TLSv1.2

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 1101 on 2024-04-10	http://www.tripwire.com/vert/?Released in ASPL 1101 on 2024-04-10

Rules

```
EXECUTE{ import aspl_ssl, dp, HIC
tls_version = 'TLSv1.2' weak_ciphers = []
text = 'TCP(%s) TLSv1.2:  ' % str
(dp.getPort())
try:  strVarName = tls_version.lower() + '_accepted_ciphers' lstCiphers = env.getContex
tVariable(strVarName) except KeyError:  rule.STOP(False) for cipher in lstCiphers:  if 'CBC' in aspl_ssl
.ssl3_cipher[cipher]:  weak_ciphers += [aspl_ssl.ssl3_cipher[cipher]] if len(weak_ciphers):  text +=
", ".join(weak_ciphers) HIC.insert_host_data_list(env.target, 'tlsv1.2_cbc_ciphers', "SSL", text) rul
e.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2024-Jul: Windows Cryptographic Services Security Feature Bypass Vulnerability	**Score**	0
Published	2024-07-09	**Strategy**	Data-Driven Attack
	nCircle: 644468	**CVSS v2**	2.4
CVSS v3	7.5		

Description

DESCRIPTION
Windows Cryptographic Services are subject to a security feature bypass vulnerability. A local attacker could bypass digital signatures upon successful exploitation of this vulnerability. Successful exploitation requires the attacker to create a SHA1 has collision.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

The patch alone does not resolve this vulnerability. The registry key HKLM\SOFTWARE\Microsoft\Cryptography\Calais\DisableCapiOverrideForRSA must also be set to 1.

Affected Applications

Application Name
Microsoft Cryptographic Services

Advisory Publisher Entries

CVE:CVE-2024-30098	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-30098
CVSSv3 Base Score: 7.5	http://www.tripwire.com/vert/cvss/?data=7.5
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I
CWE: 327	http://cwe.mitre.org/data/definitions/327.html
MSRC Guidance: CVE-2024-30098	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-30098
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1114 on 2024-07-10	http://www.tripwire.com/vert/?Released in ASPL 1114 on 2024-07-10

Rules

```
EXECUTE { import smb_file from version import Version as V, VersionException as VE from util import hexToInt
```

```
def getRegKeyValue(default_value=0):  rule.RegistryGetValue(r’HKLM\SOFTWARE\Microsoft\Cryptography\Calais\D
isableCapiOverrideForRSA’) if rule.success:  return hexToInt(rule.buffer) else:  return
default_value
try:  win_ver = env.getHostVariable( ’windows_version’ ) except KeyError:  rule.STOP( Fal
se )
def get_file_version( path, file=r’system32\ntoskrnl.exe’ ):  try:  path = r’%s\\%s’ % (path,f
ile) file_ver = smb_file.GetFileVersion(rule, None, path) ver = V(None, None, file_ver) ex
cept (VE): rule.STOP(False) return ver
try:  path = env.getHostVariable(’windows_system_root_d
irectory’) except KeyError:  rule.STOP(False)
# Vulnerable before July 2024 Patch if win_ver.startswith( ’
10.0.0.0’ ) and V( ’10.0’ ) <= get_file_version( path ) < V( ’10.0.10240.20708’ ):  rule.STOP(True) elif wi
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2024-Oct: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability I	**Score**	0
Published	2024-10-08	**Strategy**	DoS
	nCircle: 667926	**CVSS v2**	2.1
CVSS v3	7.5		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a denial of service vulnerability. An attacker could cause a denial of service condition upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
Windows Registry

Advisory Publisher Entries

CVE:CVE-2024-43483	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-43483
CVSSv3 Base Score: 7.5	http://www.tripwire.com/vert/cvss/?data=7.5
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I
CWE: 407	http://cwe.mitre.org/data/definitions/407.html
MSRC Guidance: CVE-2024-43483	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-43483
Tripwire CVSSv3 Temporal Score: 5.2	http://www.tripwire.com/vert/cvss/?data=5.2
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1127 on 2024-10-09	http://www.tripwire.com/vert/?Released in ASPL 1127 on 2024-10-09

continued on next page

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9066" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10,0.2102.1,11.0.2102,11.0.2202,11.0.2302,11.0.2402" ) THEN CAL
L isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.92
77.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302,11.0.2402" ) THEN CAL
L isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="
2.0.50727.9179" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8980" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8980" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2
.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8980" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4108.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" )
THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersio
n="4.8.4762.0" )
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.35’):  rule.STOP(True) elif V(’8.0’) <= ver < V(’
8.0.10’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2024-Oct: .NET, .NET Framework, and Visual Studio Denial of Service Vulnerability II	**Score**	0
Published	2024-10-08	**Strategy**	DoS
	nCircle: 667928	**CVSS v2**	2.1
CVSS v3	7.5		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a denial of service vulnerability. An attacker could cause a denial of service condition upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v2.x
Microsoft .NET Framework v3.0
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2022
Windows Registry

Advisory Publisher Entries

CVE:CVE-2024-43484	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2024-43484
CVSSv3 Base Score: 7.5	http://www.tripwire.com/vert/cvss/?data=7.5
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I
CWE: 407	http://cwe.mitre.org/data/definitions/407.html
MSRC Guidance: CVE-2024-43484	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2024-43484
Tripwire CVSSv3 Temporal Score: 5.2	http://www.tripwire.com/vert/cvss/?data=5.2
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1127 on 2024-10-09	http://www.tripwire.com/vert/?Released in ASPL 1127 on 2024-10-09

continued on next page

Rules

```
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9066" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10,0.2102.1,11.0.2102,11.0.2202,11.0.2302,11.0.2402" ) THEN CAL
L isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.92
77.0" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2102,11.0.2202,11.0.2302,11.0.2402" ) THEN CAL
L isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="
2.0.50727.9179" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8980" )
CALL isOSFamily( osFamily="6.0" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.dll",
startVersion="2.0.50727", patchedVersion="2.0.50727.8980" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2
.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8980" )
CALL isOSFamily( osFamily="6.0,6.1,6.2,6.3,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetV
ersion="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4108.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102,10.0.2202,10.0.2102.1,11.0.2102" )
THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersio
n="4.8.4762.0" )
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable(’.net_core_runtime’) except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V(’6.0’) <= ver < V(’6.0.35’):  rule.STOP(True) elif V(’8.0’) <= ver < V(’
8.0.10’):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith(’\\’):  path = r’%s%s’ % (path,file)
else:  path = r’%s\\%s’ % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r’HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall’, r’HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall’] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r’%s\%s\DisplayName’ % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r’%s\%s\InstallLocation’ % (uninstall_path, k) rule.Regist
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	MS-2025-Jan: .NET, .NET Framework, and Visual Studio Remote Code Execution Vulnerability	**Score**	0
Published	2025-01-14	**Strategy**	Data-Driven Attack
	nCircle: 694177	**CVSS v2**	2.4
CVSS v3	8.8		

Description

DESCRIPTION
Microsoft .NET Framework is subject to a code execution vulnerability. A local attacker could execute arbitrary code upon successful exploitation of this vulnerability.
SOLUTION
The vendor has released patches for this vulnerability. Please refer to the advisory links below.

Affected Applications

Application Name
.NET Core Runtime
Microsoft .NET Framework v3.5
Microsoft .NET Framework v4.6.x
Microsoft .NET Framework v4.7.x
Microsoft .NET Framework v4.8.1
Microsoft .NET Framework v4.8.x
Microsoft Visual Studio 2017
Microsoft Visual Studio 2019
Microsoft Visual Studio 2022
Windows Registry

Advisory Publisher Entries

CVE:CVE-2025-21176	http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-21176
CVSSv3 Base Score: 8.8	http://www.tripwire.com/vert/cvss/?data=8.8
CVSSv3 Base Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U	http://www.tripwire.com/vert/cvss/?data=CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I
MSRC Guidance: CVE-2025-21176	https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2025-21176
Tripwire CVSSv3 Temporal Score: 3.9	http://www.tripwire.com/vert/cvss/?data=3.9
Tripwire CVSSv3 Temporal Vector: (E:U/RL:O/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:O/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 1139 on 2025-01-15	http://www.tripwire.com/vert/?Released in ASPL 1139 on 2025-01-15

continued on next page

Rules

```
CALL isOSFamily( osFamily="10.0.0.0" ) THEN CALL isDotNetVulnerable(dotNetVersion="4.6", fileName="mscorlib.dl
l", startVersion="4.0.30319", patchedVersion="4.6.1953.0")
CALL isOSFamily( osFamily="10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0", fileName="mscorlib.d
ll", startVersion="2.0.50727", patchedVersion="2.0.50727.9066" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,10.0.2102.1,11.0.2202,11.0.2302,11.0.2402" ) THEN CALL isDotNet
Vulnerable( dotNetVersion="2.0", fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.
9179" )
CALL isOSFamily( osFamily="10.0.2102,10.0.2202,11.0.2202,11.0.2302.0,11.0.2402" ) THEN CALL isDotNetVulnerable
( dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9290.0" )
CALL isOSFamily( osFamily="10.0.2102.1,11.0.2302.1" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileN
ame="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.9294.0" )
CALL isOSFamily( osFamily="10.0.2202" ) THEN CALL isDotNetVulnerable( dotNetVersion="4.8", fileName="mscorlib.
dll", startVersion="4.8", patchedVersion="4.8.4772.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.0.2,10.0.1.0,10.0.0.6,10.0.2102." ) THEN CALL isDotNetVulnerable(
dotNetVersion="4.8", fileName="mscorlib.dll", startVersion="4.8", patchedVersion="4.8.4775.0" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.1.0,10.0.0.2" ) THEN CALL isDotNetVulnerable( dotNetVersion="2.0",
fileName="mscorlib.dll", startVersion="2.0.50727", patchedVersion="2.0.50727.8980" )
CALL isOSFamily( osFamily="6.1,6.2,6.3,10.0.1.0,10.0.0.2,10.0.0.6" ) THEN CALL isDotNetVulnerable( dotNetVersi
on="4.7", fileName="mscorlib.dll", startVersion="4.7", patchedVersion="4.7.4126.0" )
EXECUTE { import aspl_env from version import Version as V, VersionException as VE
try:  runtime = aspl_en
v.getContextVariable('.net_core_runtime') except KeyError:  rule.STOP(False)
for host_ver in runtime:
ver = V(host_ver) if V('8.0') <= ver < V('8.0.12'):  rule.STOP(True) elif V('9.0') <= ver < V('
9.0.1'):  rule.STOP(True)
rule.STOP(False) }
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
uninstall_pa
ths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wind
ows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.enu
mKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule.R
egistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2019" i
n rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Registry
...
EXECUTE { import util import smb_file from version import Version as V, VersionException as VE
def get_file_v
ersion(path, file):  try:  if path.endswith('\\'):  path = r'%s%s' % (path,file)
else:  path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, pa
th) ver = V(None, None, file_ver) except (VE): return None return ver
uninstall_
paths = [r'HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall', r'HKLM\SOFTWARE\wow6432node\Microsoft\Wi
ndows\CurrentVersion\Uninstall'] installDir = None
for uninstall_path in uninstall_paths:  for k in util.e
numKeys( rule, uninstall_path ):  name_path = r'%s\%s\DisplayName' % ( uninstall_path, k ) rule
.RegistryGetValue( name_path ) if rule.success and rule.buffer.startswith("Visual Studio") and " 2022"
in rule.buffer:  location = r'%s\%s\InstallLocation' % (uninstall_path, k) rule.Regist
...
EXECUTE{ import smb_file from version import Version as V, VersionException as VE
def get_file_version(path,
file): try: if path.endswith('\\'): path = r'%s%s' % (path,file) else:
path = r'%s\\%s' % (path,file)
file_ver = smb_file.GetFileVersion(rule, None, path)
ver = V(None, None, file_ver) except (VE): rule.STOP(False) return ver
path = r'H
KLM\SOFTWARE\Microsoft\VisualStudio\sxs\vs7\15.0' rule.RegistryGetValue(path) if not rule.success: rule.S
TOP(False)
ver = get_file_version(rule.buffer, file='common7\\ide\\devenv.exe')
if V('15.0') < ver < V('15.
9.35706.162'): rule.STOP(True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	X-XSS-Protection Enabled	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 507118	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The remote server headers enable X-XSS-Protection. This will activate reflected XSS protections on supported browsers. Current versions of Chrome, Firefox, and Edge do not support this header.
SOLUTION
No solution is needed. A Content-Security-Policy with unsafe-inline scripts disabled should be considered for improved protection against XSS.

Affected Applications

Application Name
HTTP

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:U/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:U/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: Released in ASPL 961 on 2021-08-24	http://www.tripwire.com/vert/?Released in ASPL 961 on 2021-08-24

Rules

```
EXECUTE { from aspl_env import getContextVariable
thisHeader = ’X_XSS_Protection’ expectedValueSubstring = ’1
’
try: Headers = getContextVariable(’HTTP_Headers’) except KeyError: rule.STOP(False)
TranscriptMess
age = ’XSS Protection header enabled for the following vhosts: ’ EnabledHosts = [] for hostname in Headers:
if thisHeader in Headers[hostname]: if not Headers[hostname][thisHeader].strip().startswith(expected
ValueSubstring): continue if hostname is None: EnabledHosts += [’(default)’]
else: EnabledHosts += [hostname]
if len(EnabledHosts) > 0: rule.transcript = Transcrip
tMessage + ’, ’.join(EnabledHosts) rule.STOP(True) rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	HTTP Available	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 1343	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The Hyper Text Transfer Protocol (HTTP) is the application level protocol used by Web servers for transferring information over the Internet.

HTTP includes several methods for web-enabled applications to interact, and is associated with specific security concerns. It is recommended that this service be enabled only on systems acting as dedicated web servers.
SOLUTION
HTTP should be disabled if it is not necessary for the planned operations of the server.

Affected Applications

Application Name
HTTP

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
STOP WITH Match
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	NetBIOS SSN Available	**Score**	0
Published		**Strategy**	Access Control Breach
	nCircle: 1492	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The NetBIOS session service (netBIOS-ssn, tcp 139) serves as a connection-oriented, reliable, sequenced transport mechanism for NetBIOS messages.

The Windows NetBIOS implementation is designed for ease-of-use with regard to network resource sharing. Windows NT/2K allows a substantial amount of information to be obtained about the network by querying NetBIOS services. There are several severe information leaks associated with default configuration of Windows NT: anonymous domain and user enumeration, share access, and remote acquisition of Registry information (a.k.a. the "Red Button" attack).
SOLUTION
We recommend the use of packet filtering on firewalls and border routers to block access to NetBIOS services of internal systems. On systems that are exposed to the Internet, entirely disable the following NetBIOS services over TCP/IP:

NetBIOS Name Service, 137/tcp and 137/udp
NetBIOS Datagram Service, 138/tcp and 138/udp
NetBIOS Session Service, 139/tcp and 139/udp

Affected Applications

Application Name
Microsoft Windows NetBIOS Session Service
NetBIOS Session Service
Samba NBSS

Advisory Publisher Entries

Sans Top 20 2001: W4	http://www.sans.org/top20/2001/?portal=738979f087d735924c39f0d8843ebedf#W4
Sans Top 20 2002: W4	http://www.sans.org/top20/2002/?portal=d545407eee69d45bca553661aa6cd41e#W4
Sans Top 20 2003: w5	http://www.sans.org/top20/2003/?portal=e4f3ca489ec98236af967652e9032da3#w5
Sans Top 20 2004: w3	http://www.sans.org/top20/2004/?portal=a9a59f93888a513a1bfa62e4af857820#w3
Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
STOP WITH Match
STOP WITH Match
STOP WITH Match
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	SMB AUTHENTICATION SUCCESS	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 5923	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
IP360 was able to log into a device, making DRT testing possible on this host.

Affected Applications

Application Name
IPv4 Layer 4
SMB-Auth

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE{ from aspl_env import getHostVariable from aspl_wdrt import ASPL_WDRT
smb_creds = rule.env.target.get
CredentialSet('SMB')
if smb_creds == []:  rule.STOP(False)
try:  host_access = getHostVariable( 'WDRT_
ACCESS' ) except KeyError:  rule.STOP( False )
if not host_access & ASPL_WDRT.WDRT_SMB_AUTH_SUCCESS: r
ule.STOP( False ) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Host has IPv6 Enabled	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 7875	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
This Windows host is capable of using IPv6 addresses, and this functionality is activated. Although the ability to process IPv6 is not currently a security vulnerability, future developments could lead to increased risk.

Affected Applications

Application Name
Host has IPv6 Enabled
Windows 2003
Windows XP

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
RegistryQuery GetKey[HKLM\SYSTEM\CurrentControlSet\Services\Tcpip6] THEN CHECK Exists
RegistryQuery GetKey[HKLM\SYSTEM\CurrentControlSet\Services\Tcpip6] THEN CHECK Exists
STOP WITH Match
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	RPC DCOM AUTHENTICATION SUCCESS	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 9971	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
RPC DCOM AUTHENTICATION SUCCESS

Affected Applications

Application Name
IPv4 Layer 4

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE{ from aspl_wmicore import ASPL_WMI from aspl_env import getHostVariable
smb_creds = rule.env.target.g
etCredentialSet('SMB')
if smb_creds == []:  rule.STOP(False)
rule = ASPL_WMI( env ) env.tls[ '__ASPL_rul
e' ] = rule
try:  host_access = getHostVariable( 'WDRT_ACCESS' ) except KeyError:  rule.STOP( False )

if not host_access & rule.WDRT_RPC_AUTH_SUCCESS: rule.STOP( False ) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	WMI AUTHENTICATION SUCCESS	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 9973	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
WMI AUTHENTICATION SUCCESS

Affected Applications

Application Name
IPv4 Layer 4

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE{ from aspl_wdrt import ASPL_WDRT from aspl_env import getHostVariable
smb_creds = rule.env.target.get
CredentialSet('SMB')
if smb_creds == []:  rule.STOP(False)
try:  host_access = getHostVariable( 'WDRT_
ACCESS' ) except KeyError:  rule.STOP( False )
if not host_access & ASPL_WDRT.WDRT_WMI_AUTH_SUCCESS: r
ule.STOP( False ) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	The contents of an SMB share may be enumerated	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 11137	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The contents of an SMB share may be enumerated, allowing users to view the files in the share.
SOLUTION
The default permissions of a Windows SMB share vary by operating system version. Ensure SMB shares have a secure access control list.

Affected Applications

Application Name
SMB-Auth
Windows Operating System

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE{ import smb_secdes, stdio, HIC from smb_file import FILE
def enumValues( key ):  rule.RegistryEnum
Values( key )
if( rule.success == False ):  return []
temp = rule.buffer.split( "\0" ) te
mp_length = len( temp ) - 1
if( temp_length > -1 and temp[ temp_length ] == "" ):  temp.pop( t
emp_length ) return temp
def enumDir( share ):  dir = FILE( rule, share, '\\' ) rule.CIFSEnumDir(
"%s:%s\\%s" % ( dir.share, dir.path, '*' ) ) if ( rule.success == False ):  return None return
rule.buffer
Shares = enumValues( "HKLM\\System\\CurrentControlSet\\Services\\LanManServer\\Shares" )
matche
d = False
for share in Shares:  if len( share ) == 0:  continue
if not enumDir( share ):
```

```
continue
matched = True HIC.insert_host_data_list( env.target, 'SMB_Shares_Which_Can_Be_Enumer
ated', 'WDRT', share ) continue
if not matched:  rule.STOP( False ) }
```

```
EXECUTE{ import smb_secdes, stdio, HIC from smb_file import FILE
try:  if env.getContextVariable( 'SMBAcc
essDenied' ):  rule.STOP( False ) except KeyError:  rule.STOP( False )
def enumShares():  rule.S
MBEnumShares()
if( rule.success == False ):  return []
temp = rule.buffer.split( '\n' ) t
emp_length = len( temp ) - 1
if( temp_length > -1 and temp[ temp_length ] == '' ):  temp.pop( temp
_length ) return temp
def enumDir( share ):  dir = FILE( rule, share, '\\' ) rule.CIFSEnumDir( "%s
:%s\\%s" % ( dir.share, dir.path, '*' ) ) if ( rule.success == False ):  return None return ru
le.buffer
shares = enumShares()
if not shares:  rule.STOP( False )
matched = False
for share in share
s:  if ( len( share ) == 0 ):  continue
if not enumDir( share ):  continue
mat
ched = True HIC.insert_host_data_list( env.target, 'SMB_Shares_Which_Can_Be_Enumerated', 'SMB', share )
continue
if not matched:  rule.STOP( False ) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	A Windows SMB share permits read access to Everyone [via SMB]	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 11144	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
A folder that grants read access to Everyone is accessible through an SMB share.
SOLUTION
The effective permissions of an SMB share are determined by the most restrictive result of the SMB permissions and the underlying file system permissions. Ensure shared folders have a secure access control list.

Affected Applications

Application Name
SMB-Auth

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE{ import smb_secdes, stdio, HIC import smb_file from dp_exceptions import SMBFailure
try:  if env.
getContextVariable( ’SMBAccessDenied’ ):  rule.STOP( False ) except KeyError:  rule.STOP( False )
d
ef enumShares():  rule.SMBEnumShares( )
if( rule.success == False ):  return []
temp = rul
e.buffer.split( ’\n’ ) temp_length = len( temp ) - 1
if( temp_length > -1 and temp[ temp_length ] ==
’’ ):  temp.pop( temp_length )
if temp.count( ’IPC$’ ):  temp.remove( ’IPC$’ )
return
temp
def getDirDacl( share ):  try:  smb_file.GetFileDACL( rule, share, ’\\’ ) except SMBFailur
e:  rule.success = False
if ( rule.success == False ):  return None return rule.buffer
```

```
shares = enumShares()
matched = False
for share in shares:  if len( share ) == 0:  continue

value = getDirDacl( share )
if not value:  continue
SecDes = smb_secdes.FileObject.UnpackSDD
...
```

Authentication Attempt

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	SSL/TLS Certificate Signature Validation Failed	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 25939	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
nCircle IP360 was unable to verify the signature on this SSL/TLS certificate. This could mean that the server will be unable to establish an encrypted tunnel.
SOLUTION
If this certificate came from a trusted certificate authority or is self-signed, then the certificate may be invalid and another should be requested.

Affected Applications

Application Name
SSL
STARTTLS Capable SMTP Server (TLSv1.0)

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
CHECK Contains[220] WITH Length[3],Offset[0] THEN SEND String[EHLO ip360.ncircle.com\x0d\x0a] THEN CHECK Conta
ins/250[ -]STARTTLS\x0d\x0a/ THEN SEND String[STARTTLS\x0d\x0a] THEN CHECK Contains[220] BEFORE Contains/Ready
to start TLS\x0d\x0a|SMTP server ready\x0d\x0a/ THEN EXECUTE{ import aspl_ssl
try:  s = aspl_ssl.newSSLSe
ssion(ssl_protocol="TLSv1") s.getServerCertificate() if not s.Server.validateCertificateSignature(pass
Exception = False):  rule.STOP(True) rule.STOP(False)
except aspl_ssl.SSLException:  rule.STOP(
False )
}
```

```
EXECUTE { try:  cert_hashes = env.getContextVariable("ssl_cert_hashes") except KeyError:  rule.STOP(Fals
e)
for cert_hash in cert_hashes:  if not cert_hashes[cert_hash]["valid_sig"]:  rule.STOP(True)
rul
e.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Untrusted SSL/TLS Certificate	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 26188	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
An SSL certificate on this host was signed by an untrusted Certificate Authority. Users that attempt to browse to a site that makes use of this certificate may be informed by the browser that the connection is untrusted, and will be forced to add an exception for the certificate in order to be able to browse the site.
SOLUTION
A certificate should be obtained from a trusted root Certificate Authority.

Affected Applications

Application Name
SSL
STARTTLS Capable SMTP Server (TLSv1.0)

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
CHECK Contains[220] WITH Length[3],Offset[0] THEN SEND String[EHLO ip360.ncircle.com\x0d\x0a] THEN CHECK Conta
ins/250[ -]STARTTLS\x0d\x0a/ THEN SEND String[STARTTLS\x0d\x0a] THEN CHECK Contains[220] BEFORE Contains/Ready
to start TLS\x0d\x0a|SMTP server ready\x0d\x0a/ THEN EXECUTE{ import aspl_ssl
try:  s = aspl_ssl.newSSLSe
ssion(ssl_protocol="TLSv1") s.getServerCertificate() rule.STOP(not s.Server.isCertificateSignedByTrust
edCA())
except aspl_ssl.SSLException:  rule.STOP( False )
}
```

```
EXECUTE { try:  cert_hashes = env.getContextVariable("ssl_cert_hashes") except KeyError:  rule.STOP(Fals
e)
for cert_hash in cert_hashes:  if not cert_hashes[cert_hash]["trusted_ca_in_chain"]:  rule.STOP(
True)
rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Microsoft Remote Desktop Service Available	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 27350	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The Microsoft Remote Desktop Service was detected on the server.

The Microsoft Remote Desktop Service (formerly known as Terminal Service) provides remote display and input capabilities over network connections for Windows-based applications running on a server. RDP is designed to support different types of network topologies and multiple LAN protocols. By default the server listens on TCP port 3389.
SOLUTION
Disable this service if it is not essential to the server's operation.

Affected Applications

Application Name
Microsoft Remote Desktop Protocol

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
STOP WITH Match
```

Hosts

Hostname	**IP Address**	**Score**
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	IP Addresses Enumerated Via NetBIOS	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 28951	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
By sending a NetBIOS query, an attacker may be able to detect all IP Addresses on a system, not just the public IP Address. This may disclose internal network information.
SOLUTION
Restrict access within a broadcast domain to trusted hosts only.

Affected Applications

Application Name
NetBIOS Name Service

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE " import HIC
dataStart = ’zp\x01\x00\x00\x01\x00\x00\x00\x00\x00\x00 ’ dataEnd = ’AA\x00\x00 \x00\x01
’ new = ’’
encodeRef = { ’A’ : ’EB’, ’B’ : ’EC’, ’C’ : ’ED’, ’D’ : ’EE’, ’E’ : ’EF’, ’F’ : ’EG’, ’G’
:  ’EH’, ’H’:’EI’, ’I’:’EJ’,’J’:’EK’, ’K’ : ’EL’, ’L’ : ’EM’, ’M’ : ’EN’, ’N’ : ’EO’, ’O’ : ’EP’, ’P’ : ’FA
’, ’Q’ : ’FB’, ’R’ : ’FC’, ’R’ : ’FC’, ’S’ : ’FD’, ’T’ : ’FE’, ’U’ : ’FF’, ’V’ : ’FG’, ’W’ : ’FH’, ’X’ : ’
FI’, ’Y’ : ’FJ’, ’Z’ : ’FK’, ’0’ :  ’DA’, ’1’ :  ’DB’, ’2’ :  ’DC’, ’3’ :  ’DD’, ’4’ :  ’DE’, ’5’ :  ’DF’, ’
6’ :  ’DG’, ’7’ :  ’DH’, ’8’ :  ’DI’, ’9’ :  ’DJ’, ’ ’ :  ’CA’, ’!’  :  ’CB’, ’\x27’ :  ’CC’, ’#’ :  ’CD’, ’\x24’
:
’CE’, ’%’ :  ’CF’, ’&’ :  ’CG’, ’\x27’ :  ’CH’, ’(’ :  ’CI’, ’)’ :  ’CJ’, ’*’ :  ’CK’, ’+’ :  ’CL’, ’,’ :  ’C
M’, ’-’ :  ’CN’, ’.’  :  ’CO’, ’=’ :  ’DN’, ’:’  :  ’DK’, ’;’ :  ’DL’, ’@’ :  ’EA’, ’^’ :  ’FO’, ’_’ :  ’FP’, ’{’ :
’HL’, ’}’ :  ’HN’, ’~’ :  ’HO’, }
def encodeName(name):  new = ’’ for char in name:  new += encod
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	Portable Storage Devices Detected (Windows)	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 47419	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
Portable storage devices are being detected (Windows).

Affected Applications

Application Name
Windows Registry

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE{
from util import enumKeys import HIC
friendlyNameList = [] deviceDescList = [] hasFriendlyNames = F
alse hasDeviceDesc = False
for path1 in enumKeys(rule, "HKLM\\SYSTEM\\CurrentControlSet\\Enum\\USB\\" ):
for path2 in enumKeys( rule, "HKLM\\SYSTEM\\CurrentControlSet\\Enum\\USB\\" + path1 ):  path3 = ("HKLM
\\SYSTEM\\CurrentControlSet\\Enum\\USB\\" + path1 + "\\" + path2) print repr(path3) rule.Regis
tryGetValue(path3 + ’\\FriendlyName’) if not rule.success:  rule.RegistryGetValue(p
ath3 + ’\\DeviceDesc’) if rule.success:  deviceDescList.append(rule.buffer) else:
friendlyNameList.append(rule.buffer)
if len(friendlyNameList) > 0:  hasFrie
ndlyNames = True if len(deviceDescList) > 0:  hasDeviceDesc = True
if hasFriendlyNames or hasDeviceDes
c:  if hasFriendlyNames:  friendlyNameString = ’Named Devices:  %s’ % str(friendlyNameList) if h
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	SSL Certificate Information	**Score**	0
Published		**Strategy**	Network Reconnaissance
	nCircle: 64658	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
SSL Certificate Information has been stored for this host. Please view the instance data or Information tab for more details.

Affected Applications

Application Name
SSL Protocol Version

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:U/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:U/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE {
ssl_cert_info = [ ('SSL Certificate Serial Number', 'ssl_cert_serialNumber'), ('SSL Signatu
re Algorithm','ssl_cert_tbsSignatureAlgorithm'), ('SSL Certificate Issuer','ssl_cert_issuer'), ('SSL C
ertificate Not Valid Before','ssl_cert_notBefore'), ('SSL Certificate Not Valid After','ssl_cert_notAfter'
), ('SSL Certificate Subject','ssl_cert_subject'), ('SSL Certificate Key Usage','ssl_cert_keyUsage'),
('SSL Certificate ext Key Usage','ssl_cert_extKeyUsage'), ('SSL Certificate MD5 Thumbprint','ssl_cert_
MD5thumbprint'), ('SSL Certificate SHA1 Thumbprint','ssl_cert_SHA1thumbprint'), ('SSL Certificate Publ
ic Key Size','ssl_cert_publicKeySize'), ] ssl_vuln_info = [ ('SSLv2 Weak Ciphers','sslv2_weak_ciphers'),
('SSLv3 Weak Ciphers','sslv3_weak_ciphers'), ('TLSv1 Weak Ciphers','tlsv1_weak_ciphers'), ('TLSv1.1
Weak Ciphers','tlsv1.1_weak_ciphers'), ('TLSv1.2 Weak Ciphers','tlsv1.2_weak_ciphers'), ]
if str(env.tar
...
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	UNRELIABLE SSL/TLS CERTIFICATE CHAIN	**Score**	0
Published		**Strategy**	Data-Driven Attack
	nCircle: 80195	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
The SSL/TLS certificate chain installed on this host is unreliable.

A digital certificate chain is a list of certificates used for authentication. The chain begins with the certificate belonging to the target service or host (certificate owner) that seeks to be authenticated followed by the certificate of some other entity who issued the previous certificate. This sequence of certificate owner (certificate subject) followed by certificate issuer continues until the end of the chain. The certificate at the end of the chain should belong to a root certificate authority (CA) that most entities on the Internet will trust. The certificates in the chain between certificate owner and root CA are called intermediate certificate authorities. For a certificate chain to be reliable, a number of criteria must be met. If one or more certificates in the chain fail one or more of these criteria, then the chain is considered to be unreliable, aka a misconfigured certificate chain.

The SSL/TLS certificate chain installed on this host is unreliable for one or more of the following reasons:

1. One or more of the certificates will not become valid until a future date; until that date, the certificate cannot be used for encryption.
2. One or more of the certificates are expired.
3. One or more of the non-root-CA certificates within the chain are self-signed.
4. One or more of the certificates have an invalid signature.
5. One or more of the certificates have been created with a weak signature algorithm.
6. One or more of the certificates utilizes a key length less than 2048 bits.
7. The certificate chain is un-ordered or the root-CA is untrusted.

SOLUTION
The certificate chain should be replaced with a properly configured certificate chain.

Affected Applications

Application Name
SSL
STARTTLS Capable SMTP Server (TLSv1.0)
STARTTLS Capable SMTP Server (SSLv3)
STARTTLS Capable SMTP Server (TLSv1.1)
STARTTLS Capable SMTP Server (TLSv1.2)

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
CHECK Contains[220] WITH Length[3],Offset[0] THEN SEND String[EHLO ip360.ncircle.com\x0d\x0a] THEN CHECK Conta
ins/250[ -]STARTTLS\x0d\x0a/ THEN SEND String[STARTTLS\x0d\x0a] THEN CHECK Contains[220] BEFORE Contains/Ready
to start TLS\x0d\x0a|SMTP server ready\x0d\x0a/ THEN EXECUTE{ import time import aspl_ssl import dp, HIC
#
Vars is_vulnerable = False
invalid = False inv = False
expired = False exp = False
self_signed = False ss =
False
weak_sig_alg = False wsa = False
trusted_ca_sig = True valid_cert_sig = True ordered_chain = True sho
rt_key = False
## Functions def isInvalid( crt ):  start_time = crt.getElement("notBefore").getData()

if start_time[-1] == "Z":  start_time = start_time[:-1]
if len(start_time) == 12:  curre
nt_time = time.strftime("%y%m%d%H%M%S", time.gmtime() ) elif len(start_time) == 14:  current_time
= time.strftime("%Y%m%d%H%M%S", time.gmtime() ) else:  msg = ('Expected notBefore with length of 12
...
```

```
CHECK Contains[220] WITH Length[3],Offset[0] THEN SEND String[EHLO ip360.ncircle.com\x0d\x0a] THEN CHECK Conta
ins/250[ -]STARTTLS\x0d\x0a/ THEN SEND String[STARTTLS\x0d\x0a] THEN CHECK Contains[220] BEFORE Contains/Ready
to start TLS\x0d\x0a|SMTP server ready\x0d\x0a/ THEN EXECUTE{ import time import aspl_ssl import dp, HIC
#
Vars is_vulnerable = False
invalid = False inv = False
expired = False exp = False
self_signed = False ss =
False
weak_sig_alg = False wsa = False
trusted_ca_sig = True valid_cert_sig = True ordered_chain = True sho
rt_key = False
## Functions def isInvalid( crt ):  start_time = crt.getElement("notBefore").getData()

if start_time[-1] == "Z":  start_time = start_time[:-1]
if len(start_time) == 12:  curre
nt_time = time.strftime("%y%m%d%H%M%S", time.gmtime() ) elif len(start_time) == 14:  current_time
= time.strftime("%Y%m%d%H%M%S", time.gmtime() ) else:  msg = ('Expected notBefore with length of 12
...
```

```
CHECK Contains[220] WITH Length[3],Offset[0] THEN SEND String[EHLO ip360.ncircle.com\x0d\x0a] THEN CHECK Conta
ins/250[ -]STARTTLS\x0d\x0a/ THEN SEND String[STARTTLS\x0d\x0a] THEN CHECK Contains[220] BEFORE Contains/Ready
to start TLS\x0d\x0a|SMTP server ready\x0d\x0a/ THEN EXECUTE{ import time import aspl_ssl import dp, HIC
#
Vars is_vulnerable = False
invalid = False inv = False
expired = False exp = False
self_signed = False ss =
False
weak_sig_alg = False wsa = False
trusted_ca_sig = True valid_cert_sig = True ordered_chain = True sho
rt_key = False
## Functions def isInvalid( crt ):  start_time = crt.getElement("notBefore").getData()

if start_time[-1] == "Z":  start_time = start_time[:-1]
if len(start_time) == 12:  curre
nt_time = time.strftime("%y%m%d%H%M%S", time.gmtime() ) elif len(start_time) == 14:  current_time
= time.strftime("%Y%m%d%H%M%S", time.gmtime() ) else:  msg = ('Expected notBefore with length of 12
...
CHECK Contains[220] WITH Length[3],Offset[0] THEN SEND String[EHLO ip360.ncircle.com\x0d\x0a] THEN CHECK Conta
ins/250[ -]STARTTLS\x0d\x0a/ THEN SEND String[STARTTLS\x0d\x0a] THEN CHECK Contains[220] BEFORE Contains/Ready
to start TLS\x0d\x0a|SMTP server ready\x0d\x0a/ THEN EXECUTE{ import time import aspl_ssl import dp, HIC
#
Vars is_vulnerable = False
invalid = False inv = False
expired = False exp = False
self_signed = False ss =
False
weak_sig_alg = False wsa = False
trusted_ca_sig = True valid_cert_sig = True ordered_chain = True sho
rt_key = False
## Functions def isInvalid( crt ):  start_time = crt.getElement("notBefore").getData()

if start_time[-1] == "Z":  start_time = start_time[:-1]
if len(start_time) == 12:  curre
nt_time = time.strftime("%y%m%d%H%M%S", time.gmtime() ) elif len(start_time) == 14:  current_time
= time.strftime("%Y%m%d%H%M%S", time.gmtime() ) else:  msg = ('Expected notBefore with length of 12
...
EXECUTE { import dp, HIC
try:  cert_hashes = env.getContextVariable("ssl_cert_hashes") except KeyError:
rule.STOP(False)
port = "TCP(" + str(dp.getPort()) + ")" text = 'The following problems have been detected
for the certificate chain provided by service on ' + port + ':  '  vulnerable = False
for cert_hash in cert_has
hes:  if cert_hashes[cert_hash]["unreliable_chain_message"]:
text += " [Certificate:  %s retrieved
with hostnames:  " % (cert_hash) for hostname in cert_hashes[cert_hash]["hostnames"]:  if ho
stname is None:  text += "<NO SNI>, " else:  text += hostname + ", "
# At least one hostname must be in the list text = text[:-2] + "]:  " + cert_hashes[cert_hash]
["unreliable_chain_message"] vulnerable = True if vulnerable:  HIC.insert_host_data_list(env.target
, 'bad_certificate_chain', "SSL", text) rule.STOP(vulnerable) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	SSL Certificate Key Length < 4096 bits	**Score**	0
Published		**Strategy**	Access Control Breach
	nCircle: 81880	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
An SSL certificate used by this host utilizes a key length less than 4096 bits.
SOLUTION
Users should generate new certificates that utilize a key length of at least 4096 bits.

Affected Applications

Application Name
SSL Protocol Version

Advisory Publisher Entries

BugTraq: 62226	http://www.securityfocus.com/bid/62226
Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE { import dp from aspl_tls_enumerator import getHostnames
try:  data = env.target.getPendingHostDat
a('ssl_cert_publicKeySize')[1] except (KeyError,TypeError):  rule.STOP(False)
for hostname in getHostnames
():  for item in data:  try:  port, item, bits = item.split() if hostname:
portcheck = 'TCP(%s:%s):'  % (hostname, dp.getPort()) else:  portcheck
= 'TCP(%s):'  % dp.getPort() if port != portcheck:  continue except ValueEr
ror:  continue try:  if int(item) < 4096:  rule.STOP(True)
except (ValueError, TypeError):  continue rule.STOP(False) }
```

Hosts

Hostname	**IP Address**	**Score**
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	SSL Certificate Key Length <= 2048 bits	**Score**	0
Published		**Strategy**	Access Control Breach
	nCircle: 81881	**CVSS v2**	0.0
CVSS v3	0.0		

Description

DESCRIPTION
An SSL certificate used by this host utilizes a key length less than or equal to 2048 bits.
SOLUTION
Users should generate new certificates that utilize a key length of more than 2048 bits.

Affected Applications

Application Name
SSL Protocol Version

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE { import dp from aspl_tls_enumerator import getHostnames
try:  data = env.target.getPendingHostDat
a(’ssl_cert_publicKeySize’)[1] except (KeyError,TypeError):  rule.STOP(False)
for hostname in getHostnames
():  for item in data:  try:  port, item, bits = item.split() if hostname:
portcheck = ’TCP(%s:%s):’  % (hostname, dp.getPort()) else:  portcheck
= ’TCP(%s):’  % dp.getPort() if port != portcheck:  continue except ValueEr
ror:  continue try:  if int(item) <= 2048:  rule.STOP(True)
except (ValueError, TypeError):  continue rule.STOP(False) }
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	SSL Certificate Key Length <= 4096 bits	**Score**	0
Published	nCircle: 81882	**Strategy**	Access Control Breach
CVSS v3	0.0	**CVSS v2**	0.0

Description

DESCRIPTION
An SSL certificate used by this host utilizes a key length less than or equal to 4096 bits.
SOLUTION
Users should generate new certificates that utilize a key length of more than 4096 bits.

Affected Applications

Application Name
SSL Protocol Version

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: No	http://www.tripwire.com/vert/?No
Tripwire: N/A	http://www.tripwire.com/vert/?N/A

Rules

```
EXECUTE { import dp from aspl_tls_enumerator import getHostnames
try:  data = env.target.getPendingHostDat
a('ssl_cert_publicKeySize')[1] except (KeyError,TypeError):  rule.STOP(False)
for hostname in getHostnames
():  for item in data:  try:  port, item, bits = item.split() if hostname:
portcheck = 'TCP(%s:%s):'  % (hostname, dp.getPort()) else:  portcheck
= 'TCP(%s):'  % dp.getPort() if port != portcheck:  continue except ValueEr
ror:  continue try:  if int(item) <= 4096:  rule.STOP(True)
except (ValueError, TypeError):  continue rule.STOP(False) }
```

Hosts

Hostname	**IP Address**	**Score**
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	BigFix	**Score**	0
Published		**Strategy**	
	Custom: 100005	**CVSS v2**	0
CVSS v3	0		

Description

Detect Bigfix

Rules

```
RegistryQuery GetKey[HKLM\SOFTWARE\BigFix\EnterpriseClient] THEN CHECK Exists
RegistryQuery GetKey[HKLM\SOFTWARE\Wow6432Node\BigFix\EnterpriseClient] THEN CHECK Exists
```

Hosts

Hostname	IP Address	Score
W6OSANADM001.myl.com	10.232.7.13	195

Vulnerability

Vulnerability Name	No UNC Paths Configured for Integrity	**Score**	0
Published	nCircle: 205862	**Strategy**	Data-Driven Attack
CVSS v3	0.0	**CVSS v2**	0.0

Description

DESCRIPTION
There are no hardened UNC paths configured in Group Policy to require the use RequireIntegrity.
SOLUTION
Configure hardened UNC paths in Group Policy to use the RequireIntegry flag as seen in http://support.microsoft.com/kb/3000483.

Affected Applications

Application Name
Windows Domain Joined Host

Advisory Publisher Entries

Tripwire CVSSv3 Temporal Score: 0.0	http://www.tripwire.com/vert/cvss/?data=0.0
Tripwire CVSSv3 Temporal Vector: (E:U/RL:W/RC:C)	http://www.tripwire.com/vert/cvss/?data=(E:U/RL:W/RC:C)
Tripwire DRT Required: Yes	http://www.tripwire.com/vert/?Yes
Tripwire: Released in ASPL 601 on 2015-02-11	http://www.tripwire.com/vert/?Released in ASPL 601 on 2015-02-11

Rules

```
EXECUTE { try:  hardened = env.getHostVariable(’hardened_unc_paths’) if len(hardened) == 0:  rul
e.STOP(True) except KeyError:  rule.STOP(False)
match = True if hardened:  for unc in hardened:
if hardened[unc][’integrity’] == 1:  match = False
rule.STOP(match) }
```

Hosts

Hostname	**IP Address**	**Score**
W6OSANADM001.myl.com	10.232.7.13	195

Applications

Service	Application	Hosts
DCE/MS RPC over TCP	DCE/MS RPC Endpoint Mapper Interface (TCP)	1
Direct SMB Hosting Service	Microsoft Windows OS Family 1809 Direct SMB Session Service	1
HTTPS	HTTP Server	1
HTTPS	HTTP-Based Application	1
HTTPS	TLSv1.2	1
IPv4 Layer 4		1
Microsoft Remote Desktop Protocol	Windows 6.x-Windows 10.x (via RDP)	1
Multi-Port Protocol	AllJoyn Router Service	1
Multi-Port Protocol	CNG Key Isolation Service	1
Multi-Port Protocol	DirectWrite	1
Multi-Port Protocol	DirectX 10.x	1
Multi-Port Protocol	DirectX 11 Build 17763	1
Multi-Port Protocol	DirectX 12 Build 17763	1
Multi-Port Protocol	DirectX 9.0c	1
Multi-Port Protocol	Google Chrome Extensions	1
Multi-Port Protocol	Google Chrome Versions	1
Multi-Port Protocol	HCL BigFix Client 10.0.7.52	1
Multi-Port Protocol	Host has IPv6 Enabled	1
Multi-Port Protocol	HTTP Service	1
Multi-Port Protocol	IKE and AuthIP IPsec Keying Modules Service	1
Multi-Port Protocol	Ink Support Feature	1
Multi-Port Protocol	IP Helper Service	1
Multi-Port Protocol	IPSec Policy Agent Service	1
Multi-Port Protocol	KDC Proxy Server Service	1
Multi-Port Protocol	Microsoft .NET Framework v4.7.x	1
Multi-Port Protocol	Microsoft Cryptographic Services	1
Multi-Port Protocol	Microsoft Internet Explorer 11	1
Multi-Port Protocol	Microsoft JET Database Engine	1
Multi-Port Protocol	Microsoft JScript	1
Multi-Port Protocol	Microsoft Korean Language IME	1
Multi-Port Protocol	Microsoft MDAC	1
Multi-Port Protocol	Microsoft Paint	1
Multi-Port Protocol	Microsoft Remote Desktop Protocol 10.0	1
Multi-Port Protocol	Microsoft SharePoint	1
Multi-Port Protocol	Microsoft SoftGrid/Application Virtualization	1
Multi-Port Protocol	Microsoft System Center Operations Monitoring Agent 2019	1
Multi-Port Protocol	Microsoft Terminal Services Client	1
Multi-Port Protocol	Microsoft VBScript	1
Multi-Port Protocol	Microsoft Visual Studio	1
Multi-Port Protocol	Microsoft Windows Server	1
Multi-Port Protocol	Microsoft Windows Telnet Client	1
Multi-Port Protocol	MPEG Layer-3 codecs	1
Multi-Port Protocol	MSXML 3.0	1
Multi-Port Protocol	MSXML 6.0	1
Multi-Port Protocol	Print Spooler Service	1

continued on next page

Service	**Application**	**Hosts**
Multi-Port Protocol	Remote Registry Service	1
Multi-Port Protocol	Smart Card Service	1
Multi-Port Protocol	SSDP Discovery Service (UPNP)	1
Multi-Port Protocol	Symantec AntiVirus	1
Multi-Port Protocol	Symantec Endpoint Protection Client	1
Multi-Port Protocol	Telephony Service	1
Multi-Port Protocol	USB Attached SCSI Protocol Service	1
Multi-Port Protocol	VMware Tools 12.4.5	1
Multi-Port Protocol	Volume Shadow Copy Service	1
Multi-Port Protocol	Windows Address Book	1
Multi-Port Protocol	Windows ATL Component	1
Multi-Port Protocol	Windows CloudExperienceHost Broker	1
Multi-Port Protocol	Windows Core Messaging	1
Multi-Port Protocol	Windows Domain Joined Host	1
Multi-Port Protocol	Windows Mail	1
Multi-Port Protocol	Windows Media Player 12	1
Multi-Port Protocol	Windows OpenSSH Client	1
Multi-Port Protocol	Windows OS (Not Server Core)	1
Multi-Port Protocol	Windows Projected File System	1
Multi-Port Protocol	Windows Remote Access Connection Manager	1
Multi-Port Protocol	Windows Remote Desktop Available	1
Multi-Port Protocol	Windows Remote Desktop Configuration Service	1
Multi-Port Protocol	Windows Script Host	1
Multi-Port Protocol	Windows Search / Windows Desktop Search	1
Multi-Port Protocol	Windows Secure Boot Enabled	1
Multi-Port Protocol	Windows Server 2019	1
Multi-Port Protocol	Windows Workstation Service	1
Multi-Port Protocol	WinSCP 6.x	1
Multi-Port Protocol	Wireless LAN AutoConfig Service Running	1
Multi-Port Protocol	WordPad	1
NetBIOS Name Service	Windows NetBIOS Name Service	1
NetBIOS Session Service	Microsoft Windows OS Family 1809 NetBIOS Session Service	1
Open TCP Port	N/A	1
Service Location Protocol (srvloc/slp) TCP		1
SMB-Auth	N/A	1
SMB-Registry	N/A	1

Audits

Network Name	Scan Profile Name	Audit Start	Audit End	Approx Hours Taken
A_AHS_Scan4_NoSIH	_Mylan: Standard Profile	04/28/2025 07:49	04/28/2025 08:05	00:16