Co-Authored by:

Product ID: AA23-074A

March 15, 2023

Threat Actors Exploit Progress Telerik Vulnerability in U.S. Government IIS Server

SUMMARY

From November 2022 through early January 2023, the Cybersecurity and Infrastructure Security Agency (CISA) and authoring organizations identified the presence of indicators of compromise (IOCs) at a federal civilian executive branch (FCEB) agency. Analysts determined that multiple cyber threat actors, including an APT actor, were able to exploit a .NET deserialization vulnerability (CVE-2019-18935) in Progress Telerik user interface (UI) for ASP.NET AJAX, located in the agency's Microsoft Internet Information Services (IIS) web server. Successful exploitation of this vulnerability allows for remote code execution. According to Progress Software, Telerik UI for ASP.NET AJAX builds before R1 2020 (2020.1.114) are vulnerable to this exploit.[1]

Actions to take today to mitigate malicious cyber activity:

- Implement a patch management solution to ensure compliance with the latest security patches.
- Validate output from patch management and vulnerability scanning against running services to check for discrepancies and account for all services.
- Limit service accounts to the minimum permissions necessary to run services.

CISA, the Federal Bureau of Investigation (FBI), and the Multi-State Information Sharing and Analysis Center (MS-ISAC) are releasing this joint Cybersecurity Advisory (CSA) to provide IT infrastructure defenders with tactics, techniques, and procedures (TTPs), IOCs, and methods to detect and protect against similar exploitation.

For a downloadable copy of IOCs, see AA23-074A.stix (STIX, 6 KB).

To report suspicious or criminal activity related to information found in this joint Cybersecurity Advisory, contact your local FBI field office or CISA's 24/7 Operations Center at Report@cisa.gov or (888) 282- 0870. When available, please include the information regarding the incident: date, time, and location of the incident; type of activity; number of people affected; type of equipment used for the activity; the name of the submitting company or organization; and a designated point of contact.

This document is marked TLP:CLEAR. Recipients may share this information without restriction. Information is subject to standard copyright rules. For more information on the Traffic Light Protocol, see cisa.gov/tlp.

TECHNICAL DETAILS

Note: This advisory uses the MITRE ATT&CK® for Enterprise framework, version 12. See the MITRE ATT&CK Tactics and Techniques section for a table of the threat actors' activity mapped to MITRE ATT&CK tactics and techniques with corresponding detection and mitigation recommendations.

Overview

CISA and authoring organizations assess that, beginning as late as November 2022, threat actors successfully exploited a .NET deserialization vulnerability (CVE-2019-18935) in an instance of Telerik UI for ASP.NET AJAX Q2 2013 SP1 (version 2013.2.717) running on an FCEB agency's Microsoft IIS server. This exploit, which results in interactive access with the web server, enabled the threat actors to successfully execute remote code on the vulnerable web server. Though the agency's vulnerability scanner had the appropriate plugin for CVE-2019-18935, it failed to detect the vulnerability due to the Telerik UI software being installed in a file path it does not typically scan. This may be the case for many software installations, as file paths widely vary depending on the organization and installation method.

In addition to CVE-2019-18935, this version (2013.2.717) of Telerik UI for ASP.NET AJAX contains the following known vulnerabilities: CVE-2017-11357, CVE-2017-11317, and CVE-2017-9248. Analysis suggests that cyber threat actors exploited CVE-2019-18935 in conjunction with either CVE-2017-11357 or CVE-2017-11317. Australian Cyber Security Centre (ACSC) Advisory 2020-004 assesses that exploitation of CVE-2019-18935 is only possible with knowledge of Telerik RadAsyncUpload encryption keys.[2] Threat actors can obtain these keys through either prior knowledge or exploitation of vulnerabilities—CVE-2017-11357 or CVE-2017-11317—present in older, unpatched versions of Telerik released between 2007 and 2017. Forensic evidence is not available to definitively confirm exploitation of either CVE-2017-11357 or CVE-2017-11317.

Threat Actor Activity

CISA and authoring organizations observed multiple cyber threat actors, including an APT actor—hereafter referred to as Threat Actor 1 (TA1)—and known cybercriminal actor XE Group—hereafter referred to as Threat Actor 2 (TA2)—conducting reconnaissance and scanning activities [T1595.002] that correlate to the successful exploitation of CVE-2019-18935 in the agency's IIS server running Telerik UI for ASP.NET AJAX [T1190].

When exploiting the vulnerability, the threat actors uploaded malicious dynamic-link library (DLL) files (some masqueraded as portable network graphics [PNG] files) [T1105] to the `C:\Windows\Temp\` directory. The malicious files were then executed from the `C:\Windows\Temp\` directory via the `w3wp.exe` process—a legitimate process that runs on IIS servers. This process is routine for handling requests sent to web servers and delivering content. The review of antivirus logs identified that some DLL files were created [T1055.001] and detected as early as August 2021.

CISA and authoring organizations confirmed that some malicious files dropped on the IIS server are consistent with a previously reported file naming convention that threat actors commonly use when exploiting CVE-2019-18935.[3] The threat actors name the files in the Unix Epoch time format and use the date and time as recorded on the target system. The file naming convention follows the

pattern `[10 digits].[7 digits].dll` (e.g., a file created on October 31, 2022, could be `1667203023.5321205.dll`).

The names of some of the PNG files were misleading. For example, file `1596835329.5015914.png`, which decodes to August 7, 2020, 21:22:09 UTC, first appeared on October 13, 2022, but the file system shows a creation date of August 7, 2020. The uncorrelated Unix Epoch time format may indicate that the threat actors used the timestomping [T1070.006] technique. This file naming convention is a primary IOC used by the threat actors.

In many cases, malicious artifacts were not available for analysis because the threat actors' malware—that looks for and removes files with the .dll file extension—removed files [T1070.004] from the `C:\Windows\Temp\` directory. Through full packet data capture analysis and reverse engineering of malicious DLL files, no indications of additional malicious activity or sub-processes were found executed by the `w3wp.exe` process. CISA and authoring organizations observed error messages being sent to the threat actors' command and control (C2) server when permission restraints prevented the service account from executing the malicious DLLs and writing new files.

Network activity analysis was consistent with the artifacts provided for review. Analysts did not observe evidence of privilege escalation or lateral movement.

Threat Actor 1

CISA and authoring organizations observed TA1 exploiting CVE-2019-18935 for system enumeration beginning in August 2022. The vulnerability allows a threat actor to upload malicious DLLs on a target system and execute them by abusing a legitimate process, e.g., the `w3wp.exe` process. In this instance, TA1 was able to upload malicious DLL files to the `C:\Windows\Temp\` directory and then achieve remote code execution, executing the DLL files via the `w3wp.exe` process.

At least nine DLL files used for discovery [TA0007], C2 [TA0011], and defense evasion [TA0005]. All of the analyzed samples have network parameters, including host name, domain name, Domain Name System (DNS) server Internet Protocol (IP) address and machine name, Network Basic Input/Output System (NetBIOS) ID, adapter information, IP address, subnet, gateway IP, and Dynamic Host Configuration Protocol (DHCP) server [T1016]. All analyzed samples communicate this collected data to a C2 server at IP address `137.184.130[.]162` or `45.77.212[.]12`. The C2 traffic to these IP addresses uses a non-application layer protocol [T1095] by leveraging Transmission Control Protocol (TCP) clear text (i.e., unencrypted) over port 443. Analysis also identified that:

- Some of the analyzed samples can load additional libraries; enumerate the system, processes, files, directories [T1083]; and write files.
- Other analyzed samples can delete DLL files ending with the `.dll` extension in the `C:\Windows\Temp\` directory on the server. TA1 may use this capability to hide additional malicious activity on the network.

CISA, in coordination with the authoring organizations, identified and observed the following threat actor IPs and timestamps associated with this activity:

Table 1: Observed TA1 IPs and Timestamps

IP Address	First Identified	Last Identified
137.184.130[.]162	09/26/2022	10/08/2022
45.77.212[.]12	10/07/2022	11/25/2022
104.225.129[.]102	10/10/2022	11/16/2022
149.28.85[.]24	10/12/2022	10/17/2022
185.186.245[.]72	10/18/2022	10/18/2022
193.8.172[.]113	09/25/2022	09/25/2022
193.8.172[.]13	09/25/2022	10/17/2022
216.120.201[.]12	10/13/2022	11/10/2022
5.34.178[.]246	09/25/2022	09/25/2022
79.133.124[.]242	09/25/2022	09/25/2022
92.38.169[.]193	09/27/2022	10/08/2022
92.38.176[.]109	09/12/2022	09/25/2022
92.38.176[.]130	09/25/2022	10/07/2022

Threat Actor 2

TA2—identified as likely the cybercriminal actor XE Group—often includes `xe[word]` nomenclature in original filenames and registered domains. Volexity lists this naming convention and other observed TTPs as common for this threat actor group.[4]

As early as August 2021, CISA and authoring organizations observed TA2 delivering malicious PNG files that, following analysis, were masqueraded DLL files to avoid detection [T1036.005]. Similar to TA1, TA2 exploited CVE-2019-18935 and was able to upload at least three unique DLL files into the `C:\Windows\Temp\` directory that TA2 executed via the `w3wp.exe` process. These DLL files drop and execute reverse (remote) shell utilities for unencrypted communication with C2 IP addresses associated with the malicious domains listed in Table 2. **Note:** At the time of analysis, the domains resolved to the listed IP addresses.

Table 2: TA2 IPs and Resolving Domains

IP Address	Resolving Domains

184.168.104[.]171	xework[.]com xegroups[.]com hivnd[.]com
144.96.103[.]245	xework[.]com

Analysis of DLL files determined the files listed in Table 3 were dropped, decoded, and attempted to connect to the respective malicious domains. Embedded payloads dropped by the DLL files were observed using the command line utility `certutil[.]exe` and writing new files as `xesvrs[.]exe` to invoke reverse shell utilities execution.

Table 3: Identified Malicious Files

Filename	Description
XEReverseShell.exe	DLL files (masqueraded as PNG files) located in the `C:\Windows\Temp\` directory contain a base64 encoded file with the internal name `XEReverseShell.exe`, which was dropped into the same directory as `sortcombat.exe`. When executed, the reverse shell utility attempts to connect to `xework[.]com` or `xegroups[.]com` to obtain the IP address of the C2 server and port number for unencrypted communication. **Note:** It is likely the threat actors changed the file extension from .dll to .png to avoid detection.
Multi-OS_ReverseShell.exe	Reverse shell utility decoded from the base64 encoded file `xesmartshell.tmp`. When executed, it will attempt to connect to `xegroups[.]com` or `xework[.]com` to obtain the IP address of the C2 server and port number for unencrypted communication.
SortVistaCompat	Base64 encoded payload dropped from `Multi-OS_ReverseShell.exe`. This file receives the C2 IP and port from `xework[.]com`.

When the TA2 malware is executed a DLL file drops an executable (`XEReverseShell.exe`) that attempts to pull a C2 IP address and port number from `xework[.]com` or `xegroups[.]com`.

- If no port or IP address is found, the program will exit.
- If a port and IP address are found, the program will establish a listener and wait for further commands.

If communication is established between the TA2 malware and the C2:

- The malware will identify the operating system (Windows or Linux) and create the appropriate shell (cmd or bash), sending system information back to the C2.
- The C2 server may send the command `xesetshell`, causing the malware to connect to the server and download a file called `small.txt`—a base64-encoded webshell that the malware decodes and places in the `C:\Windows\Temp\` directory.
- The C2 server may send the command `xequit`, causing the malware to sleep for a period of time determined by the threat actors.

The two files `xesmartshell.tmp` and `SortVistaCompat` have the capability to drop an Active Server Pages (ASPX) webshell—a base64 encoded text file small.txt decoded [T1140] as `small.aspx` [T1505.003]—to enumerate drives; to send, receive, and delete files; and to execute incoming commands. The webshell contains an interface for easily browsing files, directories, or drives on the system, and allows the user to upload or download files to any directory. No webshells were observed to be dropped on the target system, likely due to the abused service account having restrictive write permissions.

For more information on the DLLs, binaries, and webshell, see CISA MAR-10413062-1.v1 Telerik Vulnerability in U.S. Government IIS Server.

MITRE ATT&CK TACTICS AND TECHNIQUES

See Table 4 for all referenced threat actor tactics and techniques in this advisory. For assistance with mapping to the MITRE ATT&CK framework, see CISA's Decider Tool and Best Practices for MITRE ATT&CK Mapping Guide.

Table 4: Identified ATT&CK Techniques for Enterprise

Reconnaissance		
Technique Title	**ID**	**Use**
Active Scanning: Vulnerability Scanning	T1595.002	Actors were observed conducting active scanning activity for vulnerable devices and specific ports.
Initial Access		
Technique Title	**ID**	**Use**
Exploit Public-Facing Application	T1190	Actors exploited a known vulnerability in the Microsoft IIS server.

Persistence		
Technique Title	**ID**	**Use**
Server Software Component: Web Shell	T1505.003	TA2's malware dropped an ASPX webshell to enumerate drives; send, receive, and delete files; and execute commands.
Defense Evasion		
Technique Title	**ID**	**Use**
Masquerading: Match Legitimate Name or Location	T1036.005	Actors leveraged the legitimate `w3wp.exe` process on the IIS server to write malicious DLL files and evade detection.
Process Injection: DLL Injection	T1055.001	Actors loaded newly created DLLs into a running `w3wp.exe` process.
Indicator Removal: File Deletion	T1070.004	TA1's malware deleted files with ".dll" from the `C:\Windows\Temp\` directory, which may indicate hidden malicious activity on the network.
Indicator Removal: Timestomp	T1070.006	Actors modified file time attributes to insert misleading creation dates.
Decode Files	T1140	The base64 encoded text file `small.txt` decoded as the webshell `small.aspx`.
Discovery		
Technique Title	**ID**	**Use**
File and Directory Discovery	T1083	Actors enumerated the IIS server via OS fingerprinting, executed Windows processes, and collected network information. TA1's malware enumerates systems, processes, files, and directories.

System Network Configuration Discovery	T1016	TA1's malware gathers network parameters, including host name, domain name, DNS servers, NetBIOS ID, adapter information, IP address, subnet, gateway IP, and DHCP server.
Command and Control		
Technique Title	**ID**	**Use**
Ingress Tool Transfer	T1105	TA1 and TA2 uploaded malicious DLL files (some masqueraded as PNG files) to the `C:\Windows\Temp\` directory.
Non-Application Layer Protocol	T1095	Actors used a non-application layer protocol (TCP) for `w3wp.exe` process exploitation, C2, and enumeration on the IIS server.

DETECTION METHODS

CISA and authoring organizations recommend that organizations review the steps listed in this section and Table 4: Identified ATT&CK Techniques for Enterprise to detect similar activity on IIS servers.

YARA Rule

CISA developed the following YARA rule from the base proof-of-concept code for CVE-2019-18935.[5] **Note:** Authoring organizations do not guarantee all malicious DLL files (if identified) will use the same code provided in this YARA rule.

```
rule CISA_10424018_01 {
      meta:
            Author = "CISA Code & Media Analysis"
            Incident = "10424018"
            Date = "2023-02-07"
            Last_Modified = "20230216_1500"
            Actor = "n/a"
            Family = "n/a"
            Capabilities = "n/a"
            Malware_Type = "n/a"
            Tool_Type = "n/a"
            Description = "Detects open-source exploit samples"
            SHA256 = "n/a"
      strings:
            $s0 = { 3D 20 7B 20 22 63 6D 22 2C 20 22 64 2E 65 22 2C }
```

```
        $s1 = { 20 22 78 22 2C 20 22 65 22 20 7D 3B }
        $s2 = { 52 65 76 65 72 73 65 53 68 65 6C 6C 28 29 }
        $s3 = { 54 65 6C 65 72 69 6B 20 55 49 }
        $s4 = { 66 69 6C 65 6E 61 6D 65 5F 6C 6F 63 61 6C }
        $s5 = { 66 69 6C 65 6E 61 6D 65 5F 72 65 6D 6F 74 65 }
        $s6 = { 41 55 43 69 70 68 65 72 2E 65 6E 63 72 79 70 74 }
        $s7 = { 31 32 31 66 61 65 37 38 31 36 35 62 61 33 64 34 }
        $s8 = { 43 6F 6E 6E 65 63 74 53 74 61 67 69 6E 67 53 65 72 76 65 72 28
        29 }
        $s9 = { 53 74 61 67 69 6E 67 53 65 72 76 65 72 53 6F 63 6B 65 74 }
        $s10 = { 2A 62 75 66 66 65 72 20 3D 20 28 75 6E 73 69 67 6E 65 }
        $s11 = { 28 2A 29 28 29 29 62 75 66 66 65 72 3B 0A 20 20 20 20 66 75
        6E 63 28 29 3B }
        $s12 = { 75 70 6C 6F 61 64 28 70 61 79 6C 6F 61 64 28 54 65 6D 70 54
        61 72 67 65 74 }
        $s13 = { 36 32 36 31 36 66 33 37 37 35 36 66 32 66 }
    condition:
        ($s0 and $s1 and $s2) or ($s3 and $s4 and $s5 and $s6 and $s7) or ($s8
        and $s9 and $s10 and $s11) or ($s12 and $s13)
}
```

Log Collection, Retention, and Analysis

CISA, FBI, and MS-ISAC recommend that organizations utilize a centralized log collection and monitoring capability, as well as implement or increase logging and forensic data retention. Longer retention policies improve the availability of data for forensic analysis and aid thorough identification of incident scope.

- **Centralized log collection and monitoring** allows for the discovery of webshell and other exploit activity. For example, organizations should monitor for external connections made from the IIS server to unknown external IP addresses. Logging may also be available—if enabled at the router or firewall—for any outbound connections initiated with PowerShell.
- **Access- and security-focused firewall (e.g., Web Application Firewall [WAF]) logs** can be collected and stored for use in both detection and forensic analysis activities. Organizations should use a WAF to guard against publicly known web application vulnerabilities, in addition to guarding against common web application attacks.

Creation of Malicious DLLs

CISA, FBI, and MS-ISAC recommend that organizations use **process monitoring**—which provides visibility into file system and application process activity—to detect suspicious executable files running from the `C:\Windows\Temp\` directory. Process monitoring via Windows Event Code 4688 will detect the legitimate `w3wp.exe` process running suspicious DLL files and other anomalous child processes. **Note:** Enabling this event may inundate security event logging. Use centralized log collection to prevent log rollover, increase log retention and archiving, and/or enable command line event logging.

Forensic analysis commonly identified the threat actors taking the following steps:

1. Create one of the DLL files (`C:\Windows\Temp\1665890187.8690152.dll`) by process `w3wp.exe` PID 6484.
2. Load the newly created DLL into a currently running IIS process, `w3wp.exe` PID 6484.
3. Make a TCP connection using `w3wp.exe` PID 6484 to `45.77.212[.]12` over port 443.
4. Invoke `C:\Windows\System32\vcruntime140.dll` (Windows C runtime library) to execute payload.

Steps 1 and 2 occur every time a malicious DLL file is created. In some cases, an ASP .NET temp file was created, but this may have indicated benign IIS server activity. **Note:** The Process ID (PID) used in this example is unique to this investigation and is not universal. IP address `45.77.212[.]12` correlates to TA1, but the pattern can be used as general practice to identify similar activity.

Additional Searching for IIS Servers

The following information was derived from artifact analysis and is provided to equip IT infrastructure defenders searching for similar activity on an IIS server. Several artifacts can be referenced to assist in determining if CVE-2019-18935 has been successfully exploited.

File Type: DLL

Location: - %SystemDrive%\Windows\Temp

When this CVE is exploited, it uploads malicious DLL files to the `C:\Windows\Temp\` directory. The malicious DLL file naming convention translates to the exact time the file was uploaded to the server.

The time is represented in a series of digits, known as Unix Epoch time. The files observed during this investigation contained two sets of digits separated by a period (.) before the DLL extension (.dll). *Example:* `1667206973.2270932.dll`

Nearly all recovered files contain a series of 10 digits to the left of the period (.) and seven digits to the right. However, one file contained only five digits in the second set, which should be taken into consideration when writing regex patterns to search for the existence of these files. *Example Regex:* `\d{10}\.\d{1,8}\.dll`

These numbers can be copied and translated from digits into readable language with the month, day, year, hour, minute, and seconds displayed.

Log Type: IIS

Location: - %SystemDrive%\inetpub\logs\LogFiles

When investigating IIS logs, specific fields were searched for and captured during the time of each connection.

If the Unix Epoch time signature has been translated from a DLL filename, specific logs can be searched based on that time. However, if the Unix Epoch time signature has not been translated, the following will still work, but may take longer for the query to run.

The four most important fields to identify this traffic are noted in the following table. These descriptions are sourced directly from Microsoft.[6]

Table 5: Four Fields Searched in IIS Logs

General Name	Field Name	Description
Method	cs-method	Requested action; for example, a GET method
URI Stem	cs-uri-stem	Universal Resource Identifier (URI), or target, of the action
URI Query	cs-uri-query	The query, if any, that the client was trying to perform; A URI query is necessary only for dynamic pages.
Protocol Status	sc-status	Hypertext Transfer Protocol (HTTP) or File Transfer Protocol (FTP) status code

Note: Depending on how logs are collected and stored, the field names may not be an exact match; this should be taken into consideration when constructing queries.

When ingesting logs into security information and event management (SIEM), the final field names did not use a hyphen (-) but used an underscore (_).

Example: cs_method instead of cs-method

Artifacts:

Table 6: Information Contained in Two Observed IIS Events

Field Name	Artifact
cs-method	POST
cs-uri-stem	/Telerik.Web.UI.WebResource.axd
cs-uri-query	type=rau
sc-status	200 and 302

When reviewing logs, two IIS events were observed with the same timestamp each time this CVE-2019-18935 was exploited. Both events contained the same information in the cs-method, cs-uri-stem, and cs-uri-query. One event had a sc-status of 200 and the other had a sc-status of 302.

Log Type: Windows Event Application Logs

Location: -%SystemDrive%\Windows\System32\winevt\logs\Application.evtx

Kroll Artifact Parser and Extractor (KAPE), a forensic artifact collector and parser, was used to extract the Windows event logs from a backup image of the compromised IIS server. All field names refer to

the labels provided via KAPE exports. The strings are of value and can be used to locate other artifacts if different tools are used. **Note:** The payload data in the following table has been shortened to only necessary strings to obscure and protect victim information.

Table 7: Example Payload Data

Event ID	Payload
1309	3005, An unhandled exception has occurred[*redacted*]w3wp.exe[*redacted*]InvalidCastException, Unable to cast object of type 'System.Configuration.Install.AssemblyInstaller' to type 'Telerik.Web.UI.IAsyncUploadConfiguration'.\n at Telerik.Web.UI.AsyncUploadHandler.GetConfiguration(String rawData)\n at Telerik.Web.UI.AsyncUploadHandler.EnsureSetup()\n at Telerik.Web.UI.AsyncUploadHandler.ProcessRequest(HttpContext context)\n at Telerik.Web.UI.HandlerRouter.ProcessHandler(String handlerKey, HttpContext context)\n at Telerik.Web.UI.WebResource.ProcessRequest(HttpContext context)\n at System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute()\n at System.Web.HttpApplication.ExecuteStepImpl(IExecutionStep step)\n at System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously)\n\n, [*redacted*]/Telerik.Web.UI.WebResource.axd?type=rau, /Telerik.Web.UI.WebResource.axd, [*redacted*], False, [*redacted*], 15, [*redacted*], False, at Telerik.Web.UI.AsyncUploadHandler.GetConfiguration(String rawData)\n at Telerik.Web.UI.AsyncUploadHandler.EnsureSetup()\n at Telerik.Web.UI.AsyncUploadHandler.ProcessRequest(HttpContext context)\n at Telerik.Web.UI.HandlerRouter.ProcessHandler(String handlerKey, HttpContext context)\n at Telerik.Web.UI.WebResource.ProcessRequest(HttpContext context)\n at System.Web.HttpApplication.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.Execute()\n at System.Web.HttpApplication.ExecuteStepImpl(IExecutionStep step)\n at System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean& completedSynchronously)\n","Binary":""}}

Authoring organizations recommend looking for the following key strings in the payload:

- `w3wp.exe`: This is the parent process that executes the code inside the malicious DLLs.
- `System.Configuration.Install.AssemblyInstaller`: Figure 1 is from the creator's GitHub repo,[7] where the string can be observed in the code. As presented by Bishop Fox and proven during authoring organizations' investigation of IIS server logs, an exception does not mean that the exploit failed, but more likely that it executed successfully.[3]

```
"System.Configuration.Install.AssemblyInstaller",
"System.Configuration.Install",
"Version={}".format(net_version),
```

Figure 1: Threat Actor Assembly Installer

If a Werfault crash report was written, Windows event application logs may contain evidence of this—even if the DLLs have been removed from the system as part of a cleanup effort by the threat actors.

Table 8: Example Threat Actor Cleanup

EventID	ExecutableInfo	MapDescription	Payload
1000	w3wp.exe |1664175639.65719.dll |c:\windows\system32\inetsrv\w3wp.exe |C:\Windows\Temp\1664175639.65719.dll	Application Error	{"EventData":{"Data":"w3wp.exe, 8.5.9600.16384, 5215df96, 1664175639.65719.dll, 0.0.0.0, 63314d94, c00000fd, 00000000000016f8, 1708, 01d8d0a5f84af443, c:\\windows\\system32\\inetsrv\\w3wp.exe, C:\\Windows\\Temp\\1664175639.65719.dll, eed89eeb-3d68-11ed-817c-005056990ed7","Binary":""}}
1001	w3wp.exe |1664175639.65719.dll |C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe |C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe |C:\ProgramData\Microsoft\Windows\WER\ReportQueue\AppCrash_w3wp.exe	Application Crash	{"EventData":{"Data":"0, APPCRASH, Not available, 0, w3wp.exe, 8.5.9600.16384, 5215df96, 1664175639.65719.dll, 0.0.0.0, 63314d94, c00000fd, 00000000000016f8, \nC:\\Windows\\Temp\\WERE3F6.tmp.appcompat.txt\nC:\\Windows\\Temp\\WERE639.tmp.WERInternalMetadata.xml\nC:\\ProgramData\\Microsoft\\Windows\\WER\\ReportQueue\\AppCrash_w3wp.exe_d538da447d49df5862c37684118d0c25c2eff_9e3fd63b_cab_0c3ee656\\memory.hdmp\nC:\\ProgramData\\Microsoft\\Windows\\WER\\ReportQueue\\AppCrash_w3wp.exe_d538da447d49df5862c37684118d0c25c2eff_9e3fd63b_cab_0c3ee656\\triagedump.dmp, C:\\ProgramData\\Microsoft\\Windows\\WER\\ReportQueue\\AppCrash_w3wp.exe_d538

			da447d49df5862c37684118d0c25c2eff_9e3fd63b_cab_0c3ee656, 0, eed89eeb-3d68-11ed-817c-005056990ed7, 4","Binary":""}}

The EventID field maps to Windows EventIDs for an easy filter. Users can leverage the Windows EventIDs to find malicious DLL with the Unix Epoch time-based name inside the C:\Windows\Temp\ directory.

Depending how log analysis is performed, various filters can be determined. However, if regex is available, the example listed in Table 8 above can be reused to match the Unix Epoch timestamp convention to assist in filtering.

Additional Analysis

When evidence of malicious DLLs is found, reverse engineering will need to be conducted to fully understand what actions occur as the malicious files could do nearly anything. Leveraging Windows security event logs, as well as Windows PowerShell logs, may provide insight into what actions the DLLs are taking. CISA and authoring organizations recommend the following process:

1. Convert any discovered malicious DLL timestamps to readable format.
2. Export the Windows security event and PowerShell logs from the device.
    - *Default path: %SystemDrive%\Windows\System32\winevt\logs\Windows PowerShell*
    - *Default path: %SystemDrive%\Windows\System32\winevt\logs\Security.evtx*
3. Filter based on identified timestamps.
4. Search for new processes created via `w3wp.exe` in Windows security event logs (e.g., *Windows EventID 4688 New Process created*).
5. Search for new PIDs from identified events. Investigate to determine if they spawned any other processes.
    - *Example: CMD.EXE launching PowerShell or running other commands such as nslookup or netstat.* ***Note:*** *This is not an exhaustive list.*
6. Search for EventID 600 in PowerShell logs.

Trellix XDR Platform Searching

If Trellix XDR Platform is deployed in an environment and a standard HX triage audit is completed in a timely manner of the suspected use of CVE-2019-18935, an organization can search for file write events from known web processes. This will identify the executables written by the web server process. CISA and authoring organizations specifically recommend searching for the following field value pair:

Table 9: Field Value Pair for Searching

Field	Value Begins With

TextAtLowestOffset	MZ

MITIGATIONS

Note: These mitigations align with the Cross-Sector Cybersecurity Performance Goals (CPGs) developed by CISA and the National Institute of Standards and Technology (NIST). The CPGs provide a minimum set of practices and protections that CISA and NIST recommend all organizations implement. CISA and NIST based the CPGs on existing cybersecurity frameworks and guidance to protect against the most common and impactful threats, tactics, techniques, and procedures. Visit CISA's Cross-Sector Cybersecurity Performance Goals for more information on the CPGs, including additional recommended baseline protections.

Manage Vulnerabilities and Configurations

- **Upgrade all instances of Telerik UI ASP.NET AJAX to the latest version after appropriate testing.** Keep all software up to date and prioritize patching to known exploited vulnerabilities (KEVs). [CPG 5.1]
- **Prioritize remediation of vulnerabilities on internet-facing systems.** For additional guidance, see CISA Insights - Remediate Vulnerabilities for Internet-Accessible Systems. [CPG 5.1]
- **Implement a patch management solution** to ensure compliance with the latest security patches. A patch management solution that inventories all software running in addition to vulnerability scanning is recommended.
- **Ensure vulnerability scanners are configured to scan a comprehensive scope of devices and locations.** For example, as noted in the Technical Details section, the victim organization had the appropriate plugin for CVE-2019-18935, but the vulnerability went undetected due to the Telerik UI software being installed in a file path not typically scanned. To identify unpatched instances of software vulnerabilities, organizations using vulnerability scanners should be aware that all installations may not be considered "typical" and may require full file scans of web applications.
    - **Note:** Vulnerability scanners may have limitations in detecting vulnerabilities, such as only being able to identify Windows Installer-installed applications, which was the case with this agency's vulnerability scanner. The Telerik UI software was installed via a continuous integration (CI) and continuous delivery (CD) pipeline rather than the Windows Installer. This highlights the importance of using a comprehensive approach for vulnerability scanning that considers all potential installation methods and file paths.
- **Validate output from patch management and vulnerability scanning solutions against running services** to check for discrepancies and account for all services.

Segment Networks Based on Function

- **Implement network segmentation to separate network segments based on role and functionality.** Proper network segmentation significantly reduces the ability for threat actor lateral movement by controlling traffic flows between—and access to—various subnetworks.

(See CISA's Layering Network Security Through Segmentation infographic and the National Security Agency's Segment Networks and Deploy Application-Aware Defenses.) [CPG 8.1]

- **Isolate similar systems and implement micro-segmentation with granular access and policy restrictions** to modernize cybersecurity and adopt zero trust principles for both network perimeter and internal devices. Logical and physical segmentation are critical to limiting and preventing lateral movement, privilege escalation, and exfiltration. Utilize access control lists (ACLs), hardened firewalls, and network monitoring devices to regulate, monitor, and audit cross-segment access and data transfers.

Other Best Practice Mitigation Recommendations

- **Implement phishing-resistant multifactor authentication (MFA)** for as many services possible—particularly for webmail, virtual private networks (VPNs), accounts that access critical systems, and privileged accounts that manage backups.
    - MFA can still be leveraged for secure access using a jump server—an asset placed between the external and internal networks that serves as an intermediary for access—to facilitate connections if assets do not have the capability to support MFA implementation.
    - For additional guidance on secure MFA configurations, visit cisa.gov/mfa. [CPG 1.3]
- **Monitor and analyze activity logs generated from Microsoft IIS and remote PowerShell.** Collect access and security focused logs (IDS/IDPS, firewall, DLP, VPN) and ensure logs are securely stored for a specified duration informed by risk or pertinent regulatory guidance. [CPG 3.1, 3.2]
    - **Evaluate user permissions** and maintain separate user accounts for all actions and activities not associated with the administrator role, e.g., for business email, web browsing, etc. All privileges should be reevaluated on a recurring basis to validate continued need for a given set of permissions. [CPG 1.5]
- **Limit service accounts to the minimum permissions necessary to run services.** CISA observed numerous error messages in network logs indicative of failed attempts to write files to additional directories or move laterally.
- **Maintain a robust asset management policy** through comprehensive documentation of assets, tracking current version information to maintain awareness of outdated software, and mapping assets to business and critical functions.
    - Determine the need and functionality of assets that require public internet exposure. [CPG 2.3]

VALIDATE SECURITY CONTROLS

In addition to applying mitigations, CISA, FBI, and MS-ISAC recommend exercising, testing, and validating your organization's security program against the threat behaviors mapped to the MITRE ATT&CK for Enterprise framework in this advisory. CISA and co-sealers recommend testing your existing security controls inventory to assess how they perform against the ATT&CK techniques described in this advisory.

To get started:

1. Select an ATT&CK technique described in this advisory (see Table 4).
2. Align your security technologies against the selected technique.
3. Test your technologies against the technique.
4. Analyze your detection and prevention technologies' performance.
5. Repeat the process for all security technologies to obtain a set of comprehensive performance data.
6. Tune your security program—including people, processes, and technologies—based on the data generated by this process.

CISA, FBI, and MS-ISAC recommend continually testing your security program, at scale, in a production environment to ensure optimal performance against the MITRE ATT&CK techniques identified in this advisory.

RESOURCES

UNIX Timestamp Converter

REFERENCES

[1] Telerik: Exploiting .NET JavaScriptSerializer Deserialization (CVE-2019-18935)
[2] ACSC Advisory 2020-004
[3] Bishop Fox CVE-2019-18935: Remote Code Execution via Insecure Deserialization in Telerik UI
[4] Volexity Threat Research: XE Group
[5] GitHub: Proof-of-Concept Exploit for CVE-2019-18935
[6] Microsoft: Configure Logging in IIS
[7] GitHub: CVE-2019-18935

ACKNOWLEDGEMENTS

Google's Threat Analysis Group (TAG) contributed to this CSA.