TLP White

2019-126 Advisory: Vulnerable versions of Telerik UI being actively exploited by APT actor

The Australian Cyber Security Centre (ACSC) has become aware that Advanced Persistent Threat (APT) actors have been scanning for and attempting exploitation against unpatched versions of Telerik UI for ASP.NET AJAX using publically available exploits. Successful exploitation could allow an attacker to upload files to the vulnerable server to facilitate further compromise.

Details

Telerik offers a variety of products which are used to provide functionality used by web pages. In some cases, Telerik products may be installed as a third party component through other products, and as such, may be unknowingly in use.

In 2017, a security vulnerability was published that affects some Telerik products which could allow a malicious cyber actor to gain control over a server. This vulnerability is detailed in CVE-2017-9248, and similarly in CVE-2017-11317 and CVE-2017-11357. Vulnerable versions of Telerik are those published between 2007 and 2017.

Telerik issued a patch for these vulnerabilities in 2017, however due to the nature of the software, the patches may need to be manually applied.

The tools to exploit this vulnerability have been publically published and require only basic knowledge or skills to use successfully. Any servers currently running a vulnerable version should be considered at risk and remediation steps should be taken.

Recommendations

The ACSC recommends organisations consider the following actions:

Identify and patch vulnerable web servers

Identification

i) The most reliable way that agencies can identify the existence of Telerik installations is by identifying Telerik DLLs within web application root directories. As the exploitable functions within the Telerik library are located within a single DLL file, `Telerik.Web.UI.dll`, agencies can identify this file to determine Telerik usage as well as determine the product version. Identification can be performed though software asset management software or host-based inspection software. A sample PowerShell script has been provided in Appendix A – Sample PowerShell script for the detection of vulnerable DLLs to assist with identification efforts.

ii) The use of Telerik can also be performed through the inspection of Internet Information Service (IIS) web server logs and or other web application logs for known Telerik resources. The

following resources are the resources requested when using the public exploitation technique making them good candidates to search for:

- `Telerik.Web.UI.DialogHandler.aspx`
- `Telerik.Web.UI.WebResource.axd`

iii) An alternative to inspecting application logs is to implement network detection rules within network security products. A sample ruleset has been provided in Appendix B – Sample network detection rules. If Telerik is identified through log or network detection methods it is advised that agencies perform further analysis on the version of Telerik being requested to confirm if it is a vulnerable version.

iv) Network vulnerability scanners may be able to assist with the identification of Telerik within an agency, however this is probably the least reliable method of detection. If scanning for this vulnerability, please be aware that some security products such as Intrusion Prevention Systems may detect the attack and block it, leading to a false negative.

v) Once Telerik installations are identified, agencies should consult the vendor's documentation to determine if they are at risk.

Mitigate Vulerabilities

Once Telerik tools have been identified agencies should follow the vendor procedures to ensure that the risk is mitigated within the environment. Links to the vendor documentation are available in the References of this report.

Investigate for evidence of exploitation

Agencies currently using or having used vulnerable versions of Telerik should look for signs of a compromise in any environment that the Telerik product was running in.

Observed exploitation attempts have involved multiple HTTP GET and POST requests to the vulnerable resource. Looking for suspicious or patterned requests to the following resources within IIS and Application logs may reveal an attempt to exploit the vulnerability:

- `Telerik.Web.UI.WebResource.axd?type=rau`
- `Telerik.Web.UI.DialogHandler.aspx`

If multiple simultaneous requests are observed then agencies should search for the existence and/or execution of other files that may have been uploaded using this technique.

Network security products can assist in the ongoing detection of exploitation attempts. A sample ruleset has been provided in Appendix B – Sample network detection rules.

Implement complimentary security controls and/or transfer risk.

The ACSC strongly recommends the implementation of the ASD Essential 8 mitigations to mitigate threats to internet facing systems. Specifically for this vulnerability, maintaining a regular patch process and validating the application of patches reduces the risk of exploitation and is an essential part of a mature cyber program.

To limit the extent of cyber security incidents related to compromise of web servers agencies should segment and segregate internet facing servers whenever possible. Methods of network segmentation for a web server may include:

- Move the web server to an appropriate network segment (e.g. a DMZ) for the environment
- Move the web application to an externally hosted server (e.g. within a cloud hosted environment)

The following controls should be applied to externally facing servers, whether DMZ or cloud based, to limit trust and data movement into the internal network. These controls will include:

- Apply host segregation by only allowing specified communications between servers where required and over specific protocols. Additional considerations and limitations should be applied to communications between the server and network internal segments.
- Internal authentication credentials should be protected from externally facing servers. Do not use or store internal segment credentials on externally facing servers.
- An additional protection for web servers is the removal of impersonate privileges from service accounts that do not require this privilege. *Please note: This will need testing as some service accounts may require this privilege.*

Additionally; logging on externally facing servers (both operating system and application logs) should capture the appropriate events to enable a security team to effectively monitor for compromise. The logs should be centralised and continuously monitored for signs of anomalous activity.

Incident Reporting

If you have questions about this advice or have indications that your environment has been compromised, contact the ACSC by emailing **asd.assist@defence.gov.au** or calling **1300 CYBER1 (1300 292 371)**.

References

1) Relevant ACSC advisories
   a) https://www.cyber.gov.au/publications/securing-content-management-systems
   b) https://www.cyber.gov.au/publications/web-shells-threat-awareness-and-guidance
2) Telerik vulnerability advisories:
   a) https://www.telerik.com/support/kb/aspnet-ajax/details/cryptographic-weakness
   b) https://www.telerik.com/support/kb/aspnet-ajax/upload-%28async%29/details/insecure-direct-object-reference
   c) https://www.telerik.com/support/kb/aspnet-ajax/upload-%28async%29/details/unrestricted-file-upload

Traffic light protocol

Alert classification	Restriction on access and use
Red	**Highly restricted** **Access to and use by your Australian Cyber Security Centre (ACSC) contact officer(s) only.** You must ensure that your ACSC contact officer(s) does not disseminate or discuss **Red** alerts with any other person, and you shall ensure that you have appropriate systems in place to ensure that the information cannot be accessed or used by any person other than your ACSC contact officer(s).
Amber	**Restricted internal access and use only.** Subject to the below, you shall only make **Amber** alerts available to your employees on a 'needs-to-know basis' strictly for your internal purposes only to assist in the protection of your information and communications technology (ICT) systems. In some instances you may be provided with **Amber** alerts which are marked to allow you to also disclose it to your contractors or agents on a 'needs-to-know basis' strictly for your internal purposes only to assist in the protection of your ICT systems.
Green	**Restricted to closed groups and subject to confidentiality** You may share **Green** alerts with external organisations, information exchanges or individuals in the network security, information assurance or critical network infrastructure community that agree to maintain the confidentiality of the information in the alert. You may not publish or post online or otherwise release it in circumstances where confidentiality may not be maintained.
White	**Not restricted** **WHITE** alerts are not confidential. They contain information that is for public, unrestricted dissemination, publication, web-posting or broadcast. You may publish the information, subject to copyright and any restrictions or rights noted in the information.
Not classified	Any information received from the ACSC that is not classified in accordance with the Traffic light protocol must be treated as **Amber** classified unless otherwise agreed in writing by the ACSC.

Appendix A – Sample PowerShell script for the detection of vulnerable DLLs

Sample PowerShell Script

```
[CmdletBinding()]
param(
  [Parameter(Mandatory=$true)]
  [String]$searchDir
)
# Vulnerable versions listed in Burp Suite extension Telewreck.py
# Available at https://github.com/capt-meelo/Telewreck/blob/master/telewreck.py
$VULN_VERSIONS = @(
    '2007.1423', '2007.1521', '2007.1626', '2007.2918', '2007.21010',
'2007.21107', '2007.31218', '2007.31314', '2007.31425',
    '2008.1415', '2008.1515', '2008.1619', '2008.2723', '2008.2826',
'2008.21001', '2008.31105', '2008.31125', '2008.31314',
    '2009.1311', '2009.1402', '2009.1527', '2009.2701', '2009.2826',
'2009.31103', '2009.31208', '2009.31314',
    '2010.1309', '2010.1415', '2010.1519', '2010.2713', '2010.2826',
'2010.2929', '2010.31109', '2010.31215', '2010.31317',
    '2011.1315', '2011.1413', '2011.1519', '2011.2712', '2011.2915',
'2011.31115', '2011.3.1305',
    '2012.1.215', '2012.1.411', '2012.2.607', '2012.2.724', '2012.2.912',
'2012.3.1016', '2012.3.1205', '2012.3.1308',
    '2013.1.220', '2013.1.403', '2013.1.417', '2013.2.611', '2013.2.717',
'2013.3.1015', '2013.3.1114', '2013.3.1324',
    '2014.1.225', '2014.1.403', '2014.2.618', '2014.2.724', '2014.3.1024',
    '2015.1.204', '2015.1.225', '2015.1.401', '2015.2.604', '2015.2.623',
'2015.2.729', '2015.2.826', '2015.3.930', '2015.3.1111',
    '2016.1.113', '2016.1.225', '2016.2.504', '2016.2.607', '2016.3.914',
'2016.3.1018', '2016.3.1027',
    '2017.1.118', '2017.1.228', '2017.2.503', '2017.2.621', '2017.2.711',
'2017.3.913'
)
Get-ChildItem -Path $searchDir -Filter Telerik.Web.UI.dll -Recurse -ErrorAction
SilentlyContinue -Force | foreach-object {
    # In ACSC samples of the Telerik.Web.UI.dll the version number is 4 "octets"
(e.g. '2014.2.724.45'), PowerShell reports this as
"Major"."Minor"."Build"."Revision".
    #   Telewreck crafts requests using version numbers between 2 and 3 octets
long, it is assumed that all revisions are vulnerable.
    if ($_.VersionInfo.FileMajorPart -lt 2012) {
```

```
        $SimplifiedFileVersion = ($_.VersionInfo.FileVersion | Select-String -
Pattern "\d{4}\.\d{4,5}").Matches.Value
    } else {
        $SimplifiedFileVersion = ($_.VersionInfo.FileVersion | Select-String -
Pattern "\d{4}\.\d{1}\.\d{3,4}").Matches.Value
    }
    if ($VULN_VERSIONS -contains $SimplifiedFileVersion) {
        Write-Host -ForegroundColor Red "Vulnerable Telerik.Web.UI.dll identified
at '$($_.FullName)'. Version number '$($_.VersionInfo.FileVersion)' matches
version '$($SimplifiedFileVersion)' in Telewreck."
    } else {
        if ($_.VersionInfo.FileMajorPart -lt 2018) {
            Write-Host -ForegroundColor Yellow "Potentially vulnerable
Telerik.Web.UI.dll identified at '$($_.FullName)'. Version number
'$($_.VersionInfo.FileVersion)' is not included in the Telewreck vulnerable
versions, but falls within timeframe of vulnerable versions."
        } else {
            Write-Host -ForegroundColor Green "Telerik.Web.UI.dll identified at
'$($_.FullName)'. Version number '$($_.VersionInfo.FileVersion)' is not included
in the Telewreck vulnerable versions and falls outside of the vulnerability
timeframes."
        }
    }
}
```

Appendix B – Sample network detection rules

Sample Ruleset

```
alert tcp any any -> any $HTTP_PORTS (msg:"Telerik Vulnerable Versions HTTP GET";
content:"Telerik.Web.UI%2c+Version%3d20"; fast_pattern; offset: 0; depth: 500;
pcre:"/Telerik\.Web\.UI%2c\+Version%3d20(?:0(?:7\.(?:1(4(?:2[3-9]|[3-9]\d)|[5-
9]\d{2})|[2-9]\d{3}|\d{5,})|[89]\.)|1(?:[0-6]\.|7\.(?:[12]\.|3\.(?:\d{1,2}|[1-
8]\d{2}|9(?:0\d|1[0-3]))\D)))/"; reference:cve,http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-2017-9248; classtype:web-application-activity; sid:
50000001; rev: 1;)


alert tcp any $HTTP_PORTS -> any any (msg:"Telerik Vulnerable Versions HTTP
Response"; content:"Telerik.Web.UI, Version=20"; fast_pattern; offset: 0; depth:
500; pcre:"/Telerik\.Web\.UI, Version=20(?:0(?:7\.(?:1(4(?:2[3-9]|[3-9]\d)|[5-
9]\d{2})|[2-9]\d{3}|\d{5,})|[89]\.)|1(?:[0-6]\.|7\.(?:[12]\.|3\.(?:\d{1,2}|[1-
8]\d{2}|9(?:0\d|1[0-3]))\D)))/"; reference:cve,http://cve.mitre.org/cgi-
bin/cvename.cgi?name=CVE-2017-9248; classtype:web-application-activity; sid:
50000002; rev: 1;)


alert tcp any any -> any $HTTP_PORTS (msg:"Telerik Possible Encryption Key
Disclosure Attempt"; content:"GET"; offset: 0; depth: 3;
content:"/Telerik.Web.UI.DialogHandler.aspx?dp="; fast_pattern; offset: 0; depth:
500; reference:url,https://www.exploit-db.com/exploits/43873; classtype:web-
application-attack; sid: 50000003; rev: 1;)


alert tcp any any -> any $HTTP_PORTS (msg:"Telerik Possible Arbitrary File Upload
Attempt"; content:"GET"; offset: 0; depth: 3;
content:"/Telerik.Web.UI.WebResource.axd?dp="; fast_pattern; offset: 0; depth:
500; reference:url,https://www.exploit-db.com/exploits/43874; classtype:web-
application-attack; sid: 50000004; rev: 1;)


alert tcp any any -> any $HTTP_PORTS (msg:"Telerik RAU_Crypto File Upload
Exploit"; content:"POST /Telerik.Web.UI.WebResource.axd?type=rau"; fast_pattern;
offset:0; depth:45; content:"Accept-Encoding: identity"; distance:0; within:500;
content:"boundary=---------------------------68821516528156"; distance:0;
within:500; reference:url,https://www.exploit-db.com/exploits/43874;
classtype:web-application-attack; sid: 50000005; rev:1; )
```