TLP White

Advisory 2020-006: Active exploitation of vulnerability in Microsoft Internet Information Services

Document Version: 2.0

Last Updated: 22 May 2020

The Australian Signals Directorate's Australian Cyber Security Centre (ACSC) is aware that sophisticated actors are actively exploiting a deserialisation vulnerability existing in all versions of Microsoft's Internet Information Services (IIS) using the .NET framework. The vulnerability exploits the service's `VIEWSTATE` parameter to allow for remote code execution by unauthorised users.

The ACSC notes that actors have attempted to use the exploit against a number of Federal and State government agencies.

Details

The Australian Signals Directorate's Australian Cyber Security Centre (ACSC) is aware that sophisticated actors are actively exploiting a deserialisation vulnerability existing in all versions of Microsoft's Internet Information Services (IIS) using the .NET framework (.NET). The vulnerability exploits the service's `VIEWSTATE` parameter to allow for remote code execution by unauthorised users.

For actors to successfully exploit this vulnerability, they need to craft a `VIEWSTATE` parameter with malicious content. On up-to-date installs of .NET on IIS, the contents of this parameter are protected by Message Authentication Code (MAC) validation and an actor must obtain the IIS server `Machine Key` to exploit this vulnerability.

The ACSC has observed active targeting of organisations that have been previously compromised, implying that configuration files and associated keys may have been exfiltrated while the actor was present on systems running IIS.

The ACSC has also observed active targeting of organisations running other vulnerable software components, such as Telerik, that can also provide access to the required key material to perform decryption. For more information on this malicious use of Telerik, please refer to ACSC Advisory 2020-004: Targeting of Telerik CVE-2019-18935[1] and ACSC Advisory 2019-126: Vulnerable version of Telerik UI being actively exploited by APT actor.[2]

The ACSC has not observed the activity detailed in this advisory targeting Microsoft Exchange Servers however, the ACSC is aware of CVE-2020-0688 which would allow an actor to know the `Machine Key` for Microsoft Exchange Servers without gaining access to the key on the server. It is important to note that patches outlined in Microsoft advisory CVE-2020-0688[3] address the static `Machine Key` and do not mitigate the deserialisation vulnerability if the `Machine Key` becomes known.

This advisory provides indicators of the activity ACSC has observed and details proactive advice on detecting and mitigating potential exploitation of this vulnerability.

[1] ACSC Advisory 2020-004: https://www.cyber.gov.au/threats/advisory-2020-004-telerik
[2] ACSC Advisory 2019-126: https://www.cyber.gov.au/threats/advisory-2019-126
[3] Microsoft CVE-2020-0688 Security Advisory: https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-0688

TLP White

Recommendations

Detecting Compromise

The ACSC recommends examining the following sources for the described events to detect this activity. This analysis can have a high return of false positives based on the server and the websites that are being hosted. Detections should be analysed further to confirm the activity is malicious.

Please see Appendices A and B for communications samples of `VIEWSTATE` and Telerik for further information. Agencies can use these materials to detect attempts of abuse against `VIEWSTATE` and Telerik.

IIS / Reverse Proxy / Load Balancer logging

All HTTP methods where the `VIEWSTATE` parameter is set could be exploiting this vulnerability. Analysing IIS, Reverse Proxy and Load Balancer logs for HTTP requests with the following characteristics will help narrow the search and reduce false positives:

- Sequential `POST` requests receiving 500 responses.
- Excessive requests to individual files, similar to webshell detection logic.
- Large `POST` to unexpected web pages.

Networking Monitoring

Analyse network traffic for requests with the following characteristics:

- `GET` and `POST` requests with the `VIEWSTATE` parameter exceeding 2000 bytes.
- Windows Portable Executable (PE) file within the `VIEWSTATE` parameter.

IIS Debug logs

Analyse IIS debug logs for the following:

- `ViewStateException` events being generated.

Windows Event Logs

Analyse Windows Application logs looking for `Event ID 1316` with the following message and reason:

- `Event message: “Viewstate verification failed“`
- Reason: `“Viewstate was invalid”`

TLP White

Mitigation

Ensure that the MAC validation is enabled

MAC validation ensures that `VIEWSTATE` fields on web requests have not been modified.[4] This also prevents adversaries from crafting malicious payloads without having the required key material to recreate the MAC.

The ACSC recommends:

- Ensuring that you are running an up to date version of .NET on your IIS servers, particularly that the patches detailed in Microsoft Security Advisory 2905247 are applied.
- Ensuring that the following settings are configured for IIS:
  - `enableViewStateMac` is set to `'true'`
  - `aspnet:AllowInsecureDeserialization` is set to `'false'`

  Please Note: These settings may be overwritten by individual web application configurations. Organisations should confirm that no web applications are modifying the above values.
- MAC validation is on by default in up to date versions of .NET, however it can be manually turned off via a registry key. Agencies should ensure that it has not been disabled at the following registry location:
  - Location: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v<.NET Version>`
  - Key/Value: `AspNetEnforceViewStateMac = 1` (Enabled)

  Please Note: These registry key needs to be checked for all versions of .NET installed on your server.

Regenerate and replace explicit Machine Keys within Web.config

An adversary is reliant on the `Machine Keys` to pass MAC validation and subsequently leverage the `VIEWSTATE` vulnerability. If exploitation is suspected or you suspect your server `Machine Key` has been accessed, the ACSC recommends remediating `Machine Keys` by:

- Generating a new `Machine Key` or set keys to auto-generate if not in a server farm.[5]
- Ensuring that all Web Applications have unique `Machine Keys` to add another level of isolation between the applications.[6]

Encrypt plain text Machine Keys in Web.config

Encrypting the `Machine Keys` hinders an attacker's future ability to exploit the `VIEWSTATE` vulnerability if the server is compromised, or if the `Machine Keys` are leaked. Organisations should implement this by encrypting configuration sections through .NET Framework protected configuration and RSA Protected Configuration Providers.[7]

---

[4] Microsoft Resolving view state message authentication code (MAC) errors: https://support.microsoft.com/en-au/help/2915218/resolving-view-state-message-authentication-code-mac-errors

[5] Microsoft Appendix A: How to generate a <machineKey> element: https://support.microsoft.com/en-au/help/2915218/resolving-view-state-message-authentication-code-mac-errors#appendixa

[6] Microsoft Machine Key: https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831711(v%3Dws.11)

[7] Microsoft Specifying a Protected Configuration Provider: https://docs.microsoft.com/en-us/previous-versions/aspnet/68ze1hb2(v=vs.100) and Importing and Exporting Protected Configuration RSA Key Containers: https://docs.microsoft.com/en-us/previous-versions/aspnet/yxw286t2(v=vs.100)

Set the ViewState Encryption to Always (only effective for .NET 4.5+)

To ensure encryption is permanently enabled, set the `ViewStateEncryptionMode` to `Always` in the appropriate web application `web.config` file.

Please note: While this can be done in lower versions of .NET, an actor can disregard this setting by withholding the `VIEWSTATEENCRYPTED` *from the request.*

Ensure that custom error pages are in use

ASP.NET error messages may contain sensitive information that could be used to compromise the system. Organisations should implement custom error pages for ASP.NET errors to ensure that sensitive server information remains private.[8]

Other Mitigations

The ACSC recommends agencies:

- Upgrade to the latest version of .NET, ASP and IIS to ensure all available security patches are applied.
- Implement the strongest possible Crypto Routines to reduce the chance of a brute force attack against `Machine Keys.`

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).

[8] MITRE Common Weakness Enumeration, ASP.NET Misconfiguration: https://cwe.mitre.org/data/definitions/12.html

TLP White

Appendix A: Communications samples of ViewState scanning/exploitation attempts

The following samples show communications during `VIEWSTATE` scanning and exploitation attempts using proof-of-concept code. They are provided to assist organisations in the development of detection or preventative rulesets in security devices to improve the overall security posture of their networks.

ViewState scanning

```
POST / HTTP/1.1
[additional header fields removed]
__EVENTTARGET=&__EVENTARGUMENT__=&VIEWSTATEFIELDCOUNT=2&__VIEWSTATE=AAAA&__VIEWST
ATE1=AQgTNRbCgrKvLLCDQg%3D%3D&__VIEWSTATEGENERATOR=CD85D8D2&__

GET /Account/Login.aspx?__VIEWSTATE=AAAA HTTP/1.1
```

Example of IIS log __VIEWSTATE=AAAA

```
10.0.0.1 GET /Login.aspx __VIEWSTATE=AAAA 80 - 10.0.0.1
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+
Chrome/79.0.3945.88+Safari/537.36 - example.com 200 0 0 9130 634 359
185.220.100.253
```

Example of Successful exploitation IIS Log

```
10.0.0.1 POST /Login.aspx - 80 - 10.0.0.1
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+
Chrome/79.0.3945.88+Safari/537.36 - example.com 500 0 0 7821 2385 0 109.70.100.19
```

TLP White

Example of a HTTP Session of successful exploitation of an encrypted __VIEWSTATE Parameter, which writes back "AAAAAAAAAAAAAAAAAAAAAAAAAA" as a response

```
POST /Login.aspx HTTP/1.1
Host: example.com
Connection: keep-alive
Cache-Control: max-age=0
Origin: https://example.com
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/79.0.3945.88 Safari/537.36
Sec-Fetch-User: ?1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q
=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Referer: https://example.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Content-Length: 12916

------WebKitFormBoundary
Content-Disposition: form-data; name="__EVENTTARGET"


------WebKitFormBoundary
Content-Disposition: form-data; name="__EVENTARGUMENT"


------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATEFIELDCOUNT"

2
------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATE"

------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATE1"

------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATEGENERATOR"

<hex String>
------WebKitFormBoundary
Content-Disposition: form-data; name="__EVENTVALIDATION"

<base64 string>
------WebKitFormBoundary
Content-Disposition: form-data; name=" ClientState"

------WebKitFormBoundary
```

```
Content-Disposition: form-data; name=" UserName"

dd
------WebKitFormBoundary
Content-Disposition: form-data; name=" Password"

fff
------WebKitFormBoundary
Content-Disposition: form-data; name=" LoginButton"

Log In
------WebKitFormBoundary—

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=iso-8859-1
Content-Encoding: gzip
Vary: Accept-Encoding
Server: Microsoft-IIS/8.5
x-frame-options: SAMEORIGIN
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Mon, 01 Jan 2020 09:00:00 GMT
Content-Length: 122

AAAAAAAAAAAAAAAAAAAAAAAAAA
```

TLP White

Example of a HTTP Session where an encrypted __VIEWSTATE Parameter causes an internal server error

```
POST /Login.aspx HTTP/1.1
Host: example.com
Connection: keep-alive
Cache-Control: max-age=0
Origin: https://example.com
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/79.0.3945.88 Safari/537.36
Sec-Fetch-User: ?1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q
=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Referer: https://example.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Content-Length: 13600

------WebKitFormBoundary
Content-Disposition: form-data; name="__EVENTTARGET"


------WebKitFormBoundary
Content-Disposition: form-data; name="__EVENTARGUMENT"


------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATEFIELDCOUNT"

2
------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATE"

------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATE1"


------WebKitFormBoundary
Content-Disposition: form-data; name="__VIEWSTATEGENERATOR"

<Hex String>
------WebKitFormBoundary
Content-Disposition: form-data; name="__EVENTVALIDATION"

<base64 String>
------WebKitFormBoundary
Content-Disposition: form-data; name="ClientState"
```

TLP White

```
------WebKitFormBoundary
Content-Disposition: form-data; name="UserName"

dd
------WebKitFormBoundary
Content-Disposition: form-data; name="Password"

fff
------WebKitFormBoundary
Content-Disposition: form-data; name="LoginButton"

Log In
------WebKitFormBoundary--
HTTP/1.1 500 Internal Server Error
Cache-Control: private
Content-Type: text/html; charset=iso-8859-1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Mon, 01 Jan 2020 09:00:00 GMT
Content-Length: 17106

<!DOCTYPE html>
<html>
    <head>
        <title>The state information is invalid for this page and might be
corrupted.</title>
        <meta name="viewport" content="width=device-width" />
        <style>
         body {font-family:"Verdana";font-weight:normal;font-size:
.7em;color:black;}
         p {font-family:"Verdana";font-weight:normal;color:black;margin-top: -
5px}
         b {font-family:"Verdana";font-weight:bold;color:black;margin-top: -5px}
         H1 { font-family:"Verdana";font-weight:normal;font-size:18pt;color:red }
         H2 { font-family:"Verdana";font-weight:normal;font-
size:14pt;color:maroon }
         pre {font-family:"Consolas","Lucida Console",Monospace;font-
size:11pt;margin:0;padding:0.5em;line-height:14pt}
         .marker {font-weight: bold; color: black;text-decoration: none;}
         .version {color: gray;}
         .error {margin-bottom: 10px;}
         .expandable { text-decoration:underline; font-weight:bold; color:navy;
cursor:hand; }
         @media screen and (max-width: 639px) {
          pre { width: 440px; overflow: auto; white-space: pre-wrap; word-wrap:
break-word; }
         }
         @media screen and (max-width: 479px) {
          pre { width: 280px; }
         }
        </style>
    </head>

    <body bgcolor="white">
```

```
            <span><H1>Server Error in '/' Application.<hr width=100% size=1
color=silver></H1>

            <h2> <i>The state information is invalid for this page and might be
corrupted.</i> </h2></span>

            <font face="Arial, Helvetica, Geneva, SunSans-Regular, sans-serif ">

            <b> Description: </b>An unhandled exception occurred during the
execution of the current web request. Please review the stack trace for more
information about the error and where it originated in the code.

            <br><br>

            <b> Exception Details: </b>System.Web.HttpException: The state
information is invalid for this page and might be corrupted.<br><br>

            <b>Source Error:</b> <br><br>

           <table width=100% bgcolor="#ffffcc">
               <tr>
                  <td>
                      <code>

An unhandled exception was generated during the execution of the current web
request. Information regarding the origin and location of the exception can be
identified using the exception stack trace below.</code>

                  </td>
               </tr>
            </table>

            <br>

            <b>Stack Trace:</b> <br><br>

            <table width=100% bgcolor="#ffffcc">
              <tr>
                  <td>
                      <code><pre>

[ArgumentException: The serialized data is invalid.]
   System.Web.UI.ObjectStateFormatter.Deserialize(Stream inputStream) +307
   System.Web.UI.ObjectStateFormatter.Deserialize(String inputString, Purpose
purpose) +776
   System.Web.UI.Util.DeserializeWithAssert(IStateFormatter2 formatter, String
serializedState, Purpose purpose) +61
   System.Web.UI.HiddenFieldPageStatePersister.Load() +177

[ViewStateException: Invalid viewstate.
              Client IP: 10.0.0.1
              Port: 8878
              Referer: https://example.com/
              Path: /Login.aspx
```

```
            User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
            ViewState:
zWo0zr8ZnMbIwqan9akcowpsdPtYGEOORH3+luyd4r9vljPBHcqvPgI5ItgWMKiunv707khu4Fqjb1vBw
tMLjcaIg+hufC9NzHfTjkPKiny3+ycXhgH34z8/5f/sOHU2FqZHsMqnEgOfAHe6ydjwoBtnuH+d+y+8Oy
rw5pQqucJ/fDLGzPS92VhiaDVejZvrviAa9eKmF4pjFjAl+nBW71XDn2Q0bsBxT8dElnxWW8e91ALQ6QN
upBsj7Mv70Vzcf/g6rCt+7tZftUlbdh4i8WoK39Yyzd7ENcsq3XFUxvCqSLZktdzVh48gOnIylCuDwyR4
bccfG8rwVG+g+zuSnchCOs6Fm7k2qpEYc9j9nRyFjh0cGTCl8WTIhmAzwTaPWt+xrP8Euy1zrpQF8kXc6
cx696+fnYSbRuxsKC26VqnHW2Pip50po+DFvmE2vgfaky9x9H3JdmVIxr5Y8Fy7WZeJhjnEi5zU8no3Sx
XTMwRkmhbOl4DjoGENhdxV7ZuGxzwApL69PitmJMuHGWlaT8aNZjUT13x0gm5E9OpwUTSj8jJ4T6EzdjM
LJrTYqJqXVF6ByIaGkkfjY29jJBUwc7yCn5SHLoSkSxTxS0lfpteILJf+beu2g8LRezGaH2y85+nlo/NX
rt1TYqWPInuRNf8hhhfniYMKtYpKb/9kwm2bsBUwbcmZgde+UNKo7jEoXcN4X6Wrz0rYrON6P8zHs0e/n
9ccVhu9MAuahZ6HFrXCgr0d26VDwvp5HwD647EM4T2h...]

[HttpException (0x80004005): The state information is invalid for this page and
might be corrupted.]
   System.Web.UI.ViewStateException.ThrowError(Exception inner, String
persistedState, String errorPageMessage, Boolean macValidationError) +153
   System.Web.UI.HiddenFieldPageStatePersister.Load() +317
   System.Web.UI.Page.LoadPageStateFromPersistenceMedium() +367
   System.Web.UI.Page.LoadAllState() +46
   System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint,
Boolean includeStagesAfterAsyncPoint) +9458
   System.Web.UI.Page.ProcessRequest(Boolean includeStagesBeforeAsyncPoint,
Boolean includeStagesAfterAsyncPoint) +345
   System.Web.UI.Page.ProcessRequest() +75
   System.Web.UI.Page.ProcessRequest(HttpContext context) +70
   System.Web.CallHandlerExecutionStep.System.Web.HttpApplication.IExecutionStep.
Execute() +790
   System.Web.HttpApplication.ExecuteStepImpl(IExecutionStep step) +195
   System.Web.HttpApplication.ExecuteStep(IExecutionStep step, Boolean&amp;
completedSynchronously) +88
```

Appendix B: Communications samples of Telerik scanning/exploitation attempts

The following samples show communications during Telerik scanning and exploitation attempts using proof-of-concept code. They are provided to assist organisations in the development of detection or preventative rulesets in security devices to improve the overall security posture of their networks.

Crypto brute force CVE-2017-9428

```
GET
/Providers/HtmlEditorProviders/Telerik/Telerik.Web.UI.DialogHandler.aspx?dp=cwAAA
A==HTTP/1.1
Host: [host]
Accept-Encoding: identity
```

Scanning for CVE-2019-18935

```
POST /Telerik.Web.UI.WebResource.axd?type=rau HTTP/1.1
Host: [host]
Accept-Encoding: identity
Content-Length: [length]
Content-Type: multipart/form-data; boundary=--------------------------
62616f37756f2f

-----------------------------62616f37756f2f
Content-Disposition: form-data; name="rauPostData"

-----------------------------62616f37756f2f
Content-Disposition: form-data; name="file"; filename="blob"
Content-Type: application/octet-stream

test

-----------------------------62616f37756f2f
Content-Disposition: form-data; name="fileName"

RAU_crypto.bypass
-----------------------------62616f37756f2f
Content-Disposition: form-data' name="contentType"

text/html
-----------------------------62616f37756f2f
Content-Disposition: form-data; name="lastModifiedDate"

2019-01-02T03:04:05.067Z
-----------------------------62616f37756f2f
Content-Disposition: form-data; name="metdata"

{"TotalChunks":1,"ChunkIndex":0,"TotalFileSize":1,"UploadID":"testfile.txt"}
-----------------------------62616f37756f2f--
```

TLP White

Uploading a payload for CVE-2019-18935

```
POST /dotNet/noAuth/Telerik.Web.UI.WebResource.axd?type=rau HTTP/1.1
Host: [host]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:54.0) Gecko/20100101
Firefox/54.0
Accept-Encoding:gzip, deflate
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Connection: close
Accept-Language: en-US,en;q=0.5
Update-Insecure-Requests: 1
Content-Length: 95461
Content-Type: multipart/form-data; boundary=2586422672825a68efc360a82402caf0

--2586422672825a68efc360a82402caf0
Content-Disposition: form-data; name="rauPostData"

--2586422672825a68efc360a82402caf0
Content-Disposition: form-data; name="file"; filename="[Random name like
1583577119.4946656.dll]"
Content-Type: application/octet-stream
[payload]

--2586422672825a68efc360a82402caf0
Content-Disposition: form-data; name="filename"

1583577119.4946656.dll
--2586422672825a68efc360a82402caf0
Content-Disposition: form-data; name="contentType"

application/octet-stream
--2586422672825a68efc360a82402caf0
Content-Disposition: form-data; name="lastModifiedDate"

1970-01-01T00:00:00.000z
--2586422672825a68efc360a82402caf0
Content-Disposition: form-data; name="metadata"

{TotalChunks": 1, "ChunkIndex": 0, "TotalFileSize": 1, "UploadID": "[Random name
like 1583577119.4946656.dll]"}
--2586422672825a68efc360a82402caf0
```

TLP White

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.