

**Challenge:** [Malware Traffic Analysis 5 Lab](#)

**Platform:** CyberDefenders

**Category:** Network Forensics

**Difficulty:** Medium

**Tools Used:** Thunderbird, VirusTotal, Oledump, Wireshark, Zui

**Scenario:** You're working as a soc analyst at a Security Operations Center (SOC) for a Thanksgiving-themed company. One quiet evening, you hear someone knocking at the SOC analyst's entrance. As you answer the door, an exhausted mail server technician stumbles in and quickly falls to the floor. He whispers in a shaky voice, "Mail filters are down... Spam everywhere..."

As you help him up, he looks to the sky and yells, "The gates of hell have opened!" The technician immediately collapses again and softly whispers, "The horror... The horror..."

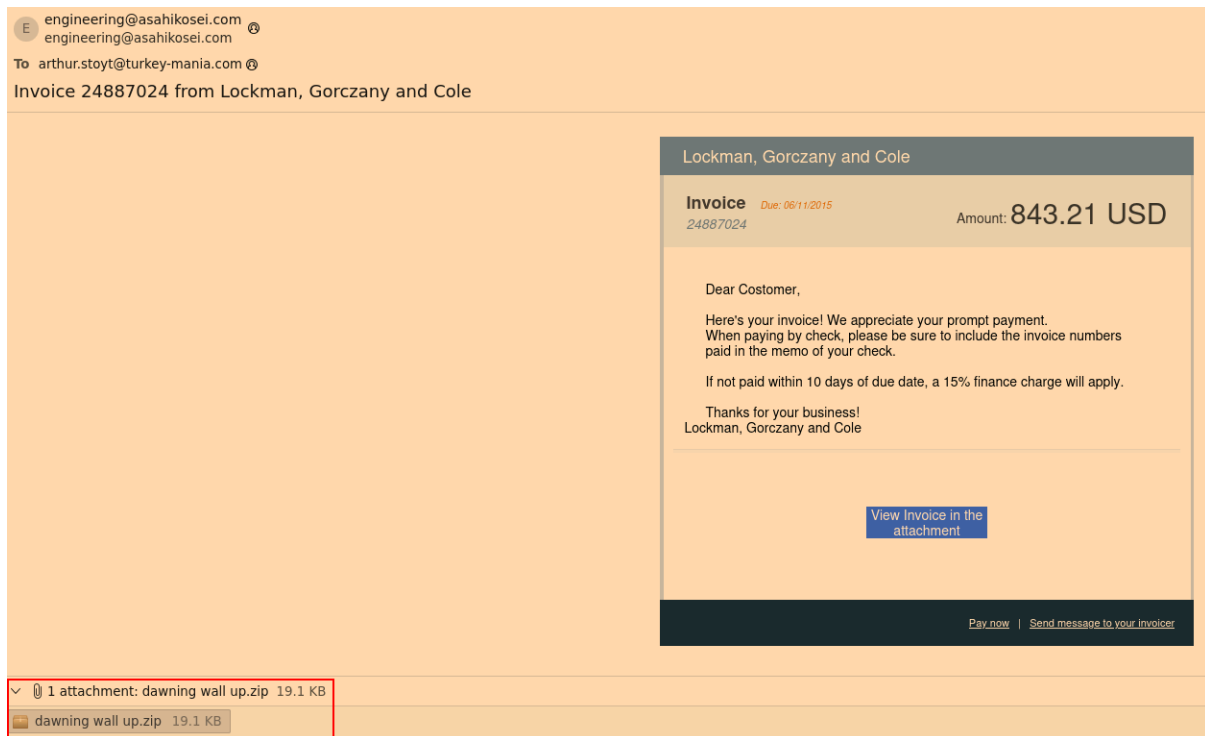
The mail filter outage lasted throughout the next day. Fortunately, very few incidents were reported. But one example caught your eye. During the mail filter outage, one of the company employees decided to play "email roulette." The employee opened one of the malicious emails from his inbox and treated it as a legitimate message.

#### **Your Assignment:**

You acquired four malicious emails the employee received. You also received a PCAP of traffic from his infected computer. Your task? Figure out which email was used to compromise the system.

#### **c41-MTA5-email-01: What is the name of the malicious file?**

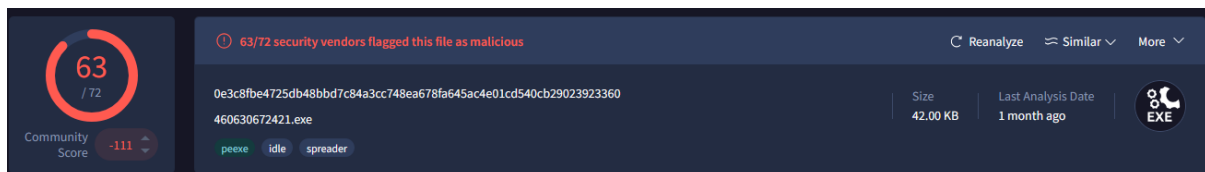
Upon opening up the .eml file in Thunerbird, we can see a supposed invoice from engineering@asahikosei.com to arthur.stoyt@turkey-mania.com containing a zip file named "dawning wall up.zip":



After saving this ZIP archive to disk, we can use the unzip command to extract its contents:

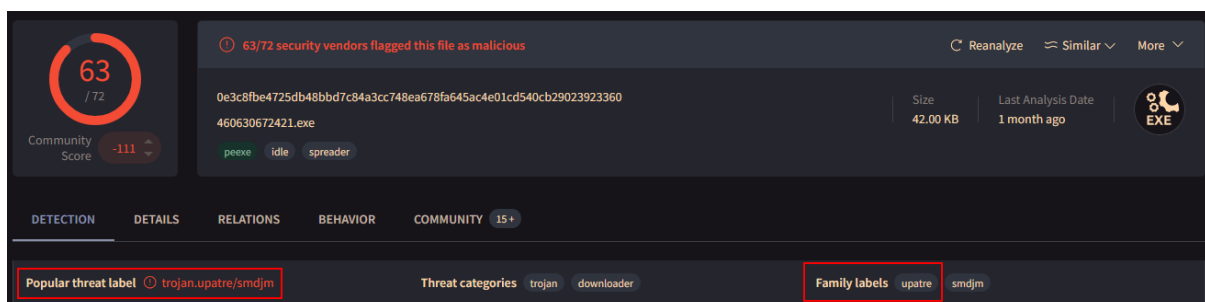
```
ubuntu@ip-172-31-22-31:~/Desktop$ unzip 'dawning wall up.zip'
Archive:  dawning wall up.zip
  inflating: 460630672421.exe
```

After running the sha256sum command against this file and submitting the hash to VirusTotal, we can see a high detection rate:



Answer: 460630672421.exe

**c41-MTA5-email-01: What is the name of the trojan family the malware belongs to? (As identified by emerging threats ruleset).**




As seen in the VirusTotal image above, this executable is given the upatre family label.

Answer: upatre

**c41-MTA5-email-02: Multiple streams contain macros in this document. Provide the number of the highest one.**

Typically, my workflow for analysing Office documents is more robust, however, given the nature of the question we can skip straight to identifying macros. Oledump is an incredible tool that enables you to find and dump macros within Office files. After opening the email:

 Mark Hoffa  
messages.5373970.157964.7c0a97a59f@messages.netsuite.com 

To Arthur Stoyt <arthur.stoyt@turkey-mania.com> 

**Payment Notification**


---

Dear Supplier,

Please find attached remittance advice for payment to be processed in your account today.

Kind Regards,  
AccountsKind Regards Macarthur Gas Pty Ltd.

---

✓  1 attachment: Bill Payment\_000010818.xls 86.0 KB

 Bill Payment\_000010818.xls 86.0 KB

You need to download the attached Excel spreadsheet. The syntax for Oledump is straightforward, all we need to supply is the filename:

```

ubuntu@ip-172-31-22-31:~/Desktop$ python3 ~/Desktop/Start\ here/Tools/DidierStevensSuite/oledump.py 'Bill Payment_000010818.xls'
/home/ubuntu/Desktop/Start here/Tools/DidierStevensSuite/oledump.py:188: SyntaxWarning: invalid escape sequence '\D'
manual = ''
1: 104 '\x01CompObj'
2: 236 '\x05DocumentSummaryInformation'
3: 216 '\x05SummaryInformation'
4: 13218 'Workbook'
5: 615 'VBA PROJECT CUR/PROJECT'
6: 131 'VBA PROJECT CUR/PROJECTwm'
7: M 24051 'VBA PROJECT CUR/VBA/Module1'
8: M 25828 'VBA PROJECT CUR/VBA/Module2'
9: 5853 'VBA PROJECT CUR/VBA/VBA PROJECT'
10: 2278 'VBA PROJECT CUR/VBA/_SRP_0'
11: 642 'VBA PROJECT CUR/VBA/_SRP_1'
12: 1244 'VBA PROJECT CUR/VBA/_SRP_2'
13: 264 'VBA PROJECT CUR/VBA/_SRP_3'
14: 812 'VBA PROJECT CUR/VBA/_SRP_4'
15: 204 'VBA PROJECT CUR/VBA/_SRP_5'
16: 622 'VBA PROJECT CUR/VBA/dir'
17: m 992 'VBA PROJECT CUR/VBA/Лист1'
18: m 992 'VBA PROJECT CUR/VBA/Лист2'
19: m 992 'VBA PROJECT CUR/VBA/Лист3'
20: M 1458 'VBA PROJECT CUR/VBA/ЭтаКнига'

```

Streams with a capital M contain macros. In this case, the highest stream that contains a macro is 20.

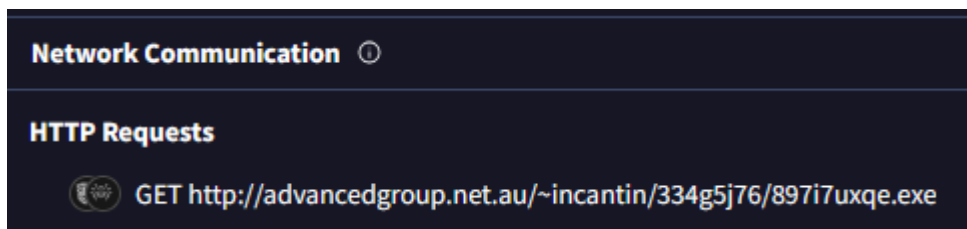
Answer: 20

**c41-MTA5-email-02: The Excel macro tried to download a file. Provide the full URL of this file?**

To see what network connections this Excel macro makes, let's hash the file and submit it to VirusTotal:

- sha256sum 'Bill Payment\_000010818.xls'

If you navigate to the Network Communication section under the Behaviour tab, we can see that it makes one GET request to download an executable called 897i7uxqe.exe:



Answer: <http://advancedgroup.net.au/~incantin/334g5j76/897i7uxqe.exe>

**c41-MTA5-email-02: The Excel macro writes a file to the temp folder. Provide the filename?**

Within the Behaviour tab of VirusTotal, if you go to the Files Written section, we can see what files this macro wrote to the temp folder:

%TEMP%\tghttp.exe

We can see that it saved tghttp.exe to the TEMP folder.

Answer: tghtop.exe

**c41-MTA5-email-03: Provide the FQDN used by the attacker to store the login credentials?**

American Express Alerts  
AMEXPGNEUSCN0006006@verizon.net

## Important Information About Your Card Membership!

### Important Information About Your Card Membership

Dear Customer,

Please note that we have introduced a new online authentication procedures in order to protect the private information of our customers.

You are required to confirm your online details with us as you will not be able to have access to your accounts until this has been done.

**Kindly open the attachment to confirm your online details.**

Once you've completed this you'll be able to manage your money whenever you want, giving you more control of your finances.

Sincerely,

American Express Customer Care

**Intended For your security:**

1 attachment: AmericanExpress.html 106 KB

AmericanExpress.html 106 KB

This email contains a .html file as an attachment. Let's save this file and explore the code using mousepad. I eventually came across mentions of jpm motos.pt.

Answer: jpm motos.pt

**c41-MTA5-email-04: How many FQDNs are present in the malicious js?**

After extracting the zip archive attachment, we can see a .js file:

```
ubuntu@ip-172-31-21-66:~/Desktop$ unzip fax000497762.zip
Archive:  fax000497762.zip
  inflating: fax000497762.doc.js
```

If you view the JavaScript, we can see some obfuscated JavaScript string concatenations. To deobfuscate the script, I contracted the help of ChatGPT which extracted IOCs:

#### What each piece does

- Domains tried: `kennedy.sitoserver.com`, `nzvincent.com`, `abama.org`
- URL path used: `/counter/?id=<HEX>&rnd=309034<n>` where `<HEX>` is your `str` value and `<n>` is 1→3.
- Download method: `MSXML2.XMLHTTP` (synchronous GET) → `ADODB.Stream` (binary write).
- Drop locations: `%TEMP%\799751.exe`, `%TEMP%\799752.exe`, `%TEMP%\799753.exe`.
- Execution: `WScript.Shell.Run(..., 2, true)` → runs hidden and (likely) waits.
- Obfuscation tricks: split-assembled string fragments, `ExpandEnvironmentStrings("%TEMP%")`, path separator via `String.fromCharCode(92)` (`\`), and broad `try/catch` to stay quiet on failure.

#### IOCs to pull/block

- Domains:
  - `kennedy.sitoserver.com`
  - `nzvincent.com`
  - `abama.org`
- Paths / pattern: `http://<domain>/counter/?id=<hex>&rnd=309034[1-3]`
- Files: `%TEMP%\79975[1-3].exe`
- Process lineage: `wscript.exe` / `cscript.exe` spawning those EXEs (hidden window style = 2).

If you want, paste me the fetched EXE (or its hash / a hexdump snippet) and I'll help you triage what it is (packer, imports, strings, behaviour) and build YARA/Sigma around these artefacts.

Answer: 3

**c41-MTA5-email-04: What is the name of the object used to handle and read files?**

The script uses `ADODB.Stream` to handle/read/write the downloaded file:

```
var xa = WScript.CreateObject("ADODB.Stream");
xa.open();
xa.type = 1; // 1 = binary, 2 = text
xa.write(xo.responseBody); // write HTTP response body into the stream
xa.position = 0; // rewind before saving
xa.saveToFile(fn + ".exe", 2); // 2 = overwrite if exists
xa.close();
```

Answer: ADODB.Stream

**c41-MTA5.pcap: The victim received multiple emails; however, the user opened a single attachment. Provide the attachment filename.**

To file the attachment filename, let's start by viewing the notice logs within Zui:

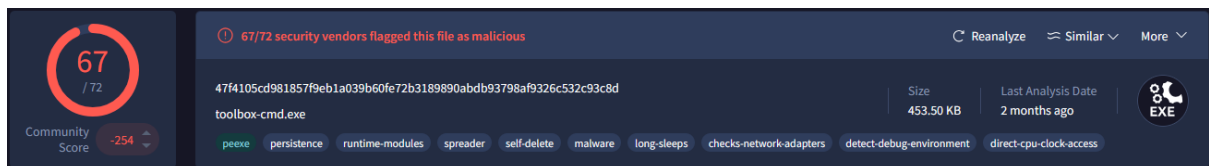
- `_path=="notice"`

We can see that an executable was downloaded from

<http://kennedy.sitoserver.com/counter/?id=5552505E160B0601161017241605070F17140507014A070B095E3C5E060A1E4A070B094A091D5E17555E555050525C50505555505E55&rnd=3090343>:

```
ts: 2015-11-06T22:22:55.991374Z,  
uid: "CvenGk44oDwZUeQhB3",  
id: {  
  orig_h: 10.3.66.103,  
  orig_p: 49158 (port=(uint16)),  
  resp_h: 174.121.246.162,  
  resp_p: 80 (port=(uint16))  
},  
fuid: "F0SuIc4zp13EQmUvB",  
file_mime_type: "application/x-dosexec",  
file_desc: "http://kennedy.sitoserver.com/counter/?id=5552505E160B0601161017241605070F17140507014A070B095E3C5E060A1E4A070B094A091D5E17555E555050525C50505555505E55&rnd=3090343",  
proto: "tcp" (zenum),  
note: "TeamCymruMalwareHashRegistry::Match" (zenum),  
msg: "Malware Hash Registry Detection rate: 51% Last seen: 2021-12-03 11:52:57",  
sub: "https://www.virustotal.com/gui/search/ce1f0b7dfd91fec1dd0b9a539f7a2c12f2be39b2",  
src: 10.3.66.103,  
dst: 174.121.246.162,  
p: 80 (port=(uint16)),
```

If you visit the provided VirusTotal link, we can see that it receives a significant number of detections:



If you recall in question 7, we found [kennedy.sitoserver.com](http://kennedy.sitoserver.com) within the obfuscated JavaScript code. This matches the domain associated with the malicious executable download found within the notice logs.

Answer: fax000497762.zip

**c41-MTA5.pcap: What is the IP address of the victim machine?**

If you look at the source host which downloaded the malicious executables from [kennedy.sitoserver.com](http://kennedy.sitoserver.com), we can find the IP address of the victim machine:

```
ts: 2015-11-06T22:22:55.991374Z,  
uid: "CvenGk44oDwZUeQhB3",  
id: {  
  orig_h: 10.3.66.103,  
  orig_p: 49158 (port=(uint16)),
```

Alternatively, if you navigate to Statistics > Conversations > IPv4, we can see that 10.3.66.103 is observed within every conversation:

Ethernet · 4	IPv4 · 321	IPv6	TCP · 389	UDP · 448					
Address A	Address B	Packets →	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
10.3.66.103	148.251.80.172	2,557	2 MB	1,265	1,000 kB	1,292	1 MB	116.062424	242.1391
10.3.66.103	109.68.191.31	1,865	1 MB	943	58 kB	922	1 MB	119.027210	7.5238
10.3.66.103	174.121.246.162	1,394	1 MB	701	43 kB	693	978 kB	104.250857	3.6402
10.3.66.103	8.8.8.8	838	97 kB	419	36 kB	419	61 kB	108.997343	247.6913
10.3.66.103	192.241.179.166	670	476 kB	296	41 kB	374	435 kB	297.378976	59.3096
10.3.66.103	74.125.226.176	466	299 kB	226	22 kB	240	278 kB	112.866562	229.6544
10.3.66.103	23.218.210.155	416	266 kB	202	15 kB	214	250 kB	111.438098	204.8735
10.3.66.103	10.3.66.1	273	37 kB	50	5 kB	223	32 kB	3.467505	337.4266
10.3.66.103	93.184.215.200	122	67 kB	63	5 kB	59	63 kB	231.167531	120.0017
10.3.66.103	205.185.216.10	105	79 kB	40	4 kB	65	75 kB	309.062144	29.5544
10.3.66.103	54.201.30.58	103	13 kB	58	5 kB	45	7 kB	114.374933	235.9653
10.3.66.103	217.160.165.207	102	14 kB	58	5 kB	44	9 kB	115.569059	232.8630
10.3.66.103	10.3.66.255	90	11 kB	90	11 kB	0	0 bytes	3.225939	336.5480
10.3.66.103	23.78.253.223	89	66 kB	37	5 kB	52	61 kB	339.525900	13.4770
10.3.66.103	94.31.29.43	77	37 kB	34	4 kB	43	33 kB	301.866600	29.5690
10.3.66.103	204.79.197.200	70	32 kB	35	3 kB	35	29 kB	266.085421	85.0453
10.3.66.103	172.226.103.54	66	52 kB	28	3 kB	38	49 kB	338.500721	13.7534
10.3.66.103	74.125.226.185	63	55 kB	23	2 kB	40	53 kB	313.154765	1.1834
10.3.66.103	69.172.216.55	61	35 kB	26	3 kB	35	32 kB	339.527366	8.0421
10.3.66.103	184.51.144.120	58	48 kB	22	2 kB	36	46 kB	341.369694	0.1559
10.3.66.103	23.72.255.183	54	36 kB	26	2 kB	28	34 kB	340.896508	0.7895
10.3.66.103	173.239.36.121	52	9 kB	25	3 kB	27	7 kB	290.561783	50.2936
10.3.66.103	173.239.42.219	50	12 kB	24	3 kB	26	9 kB	292.359011	49.6052
10.3.66.103	23.220.148.51	42	28 kB	18	2 kB	24	26 kB	352.556353	1.7293
10.3.66.103	96.16.134.180	42	27 kB	20	3 kB	22	24 kB	339.526156	5.0058
10.3.66.103	74.125.139.95	41	7 kB	22	3 kB	19	4 kB	301.865329	52.7441
10.3.66.103	199.16.172.81	35	13 kB	16	2 kB	19	11 kB	308.721952	44.3897
10.3.66.103	74.125.226.13	33	26 kB	13	1 kB	20	24 kB	353.358440	1.3675
10.3.66.103	54.148.180.204	32	7 kB	21	3 kB	11	4 kB	115.280941	230.5146
10.3.66.103	74.125.226.162	32	8 kB	21	3 kB	11	5 kB	112.380707	229.4992
10.3.66.103	74.125.226.161	29	16 kB	14	3 kB	15	13 kB	304.545648	9.9245
10.3.66.103	74.125.226.186	28	18 kB	12	2 kB	16	16 kB	341.370352	6.1608
10.3.66.103	31.192.112.238	27	4 kB	16	3 kB	11	1 kB	282.178558	46.3464
10.3.66.103	173.239.36.117	23	2 kB	12	1 kB	11	803 bytes	289.409386	45.7019
10.3.66.103	93.190.142.64	21	5 kB	12	3 kB	9	2 kB	258.613332	92.6446
10.3.66.103	93.190.141.180	18	2 kB	10	934 bytes	8	665 bytes	201.563281	120.2917
10.3.66.103	8.18.45.68	17	3 kB	11	2 kB	6	2 kB	357.298684	0.9025
10.3.66.103	74.125.139.154	16	6 kB	9	1 kB	7	5 kB	326.784762	3.5353

Answer: 10.3.66.103

**c41-MTA5.pcap: What is the FQDN that hosted the malware?**

We discovered the FQDN that hosted the malware previously as kennedy.sitoserver.com.

Answer: kennedy.sitoserver.com

**c41-MTA5.pcap: The opened attachment wrote multiple files to the TEMP folder. Provide the name of the first file written to the disk?**

If you look at the deobfuscated JavaScript, you can see that the first file written to the TEMP folder is 7997551.exe.

Answer: 7997551.exe

**c41-MTA5.pcap: One of the written files to the disk has the following md5 hash "35a09d67bee10c6aff48826717680c1c"; Which registry key does this malware check for its existence?**



Using the following query in Zui, we can pinpoint the source of this file:

- `_path=="files" md5=="35a09d67bee10c6aff48826717680c1c" | cut id.orig_h, id.resp_h, source, filename, md5`

id	source	filename	md5
> {orig_h: 10.3.66.103, resp_h: 174.121.246.162}	HTTP	2b1ce0b83.gif	35a09d67bee10c6aff48826717680c1c

If you navigate to File > Export Objects > HTTP, we can export the file for manually analysis:

1480      kennedy.sitosever.com      image/gif      464 kB

If you run strings against this file, we can find an interesting string that appears to be a registry key path:

```
interface\{9a83a958-b859-11d1-aa90-00aa00ba3258}
```

Answer: 9a83a958-b859-11d1-aa90-00aa00ba3258

**c41-MTA5.pcap: One of the written files to the disk has the following md5 hash "e2fc96114e61288fc413118327c76d93" sent an HTTP post request to "upload.php" page. Provide the webserver IP. (IP is not in PCAP)**

This is another file downloaded from kennedy.sitosever.com as observed in the output of the following Zui query:

- `_path=="files" md5=="e2fc96114e61288fc413118327c76d93" | cut id.orig_h, id.resp_h, source, filename, md5`

id	source	filename	md5
> {orig_h: 10.3.66.103, resp_h: 174.121.246.162}	HTTP	a340de8dc.gif	e2fc96114e61288fc413118327c76d93

If you upload the hash to VirusTotal, we can see that it contacts 78.24.220.229 over HTTP:

```
TCP 78.24.220.229:80
```

Answer: 78.24.220.229

**c41-MTA5.pcap: The malware initiated callback traffic after the infection. Provide the IP of the destination server.**

If you navigate to Statistics > Conversations > IPv4, we can see a large number of packets between our victim host and 109.68.191.31 over a short period of time:

Ethernet · 4	IPv4 · 321	IPv6	TCP · 389	UDP · 448						
Address A	Address B	Packets ▲	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	
10.3.66.103	148.251.80.172	2,557	2 MB	1,265	1,000 kB	1,292	1 MB	116.062424	242.1391	
10.3.66.103	109.68.191.31	1,865	1 MB	943	58 kB	922	1 MB	119.027210	7.5238	

Answer: 109.68.191.31