Spearphishing-APT

Threat hunting is working with the realization that the attacker has already compromised the business network and we are looking for signs of footprints. By referencing SANS, think in terms of who, what, where, when and why. Threat hunting builds upon the kill chain model, MIITRE ATT&CK which has various pathways(over 150).

How does one start, well it’s a series of failed experiments and successful ones. Start broadly and narrow on specific time ranges. Stay on a specific technique, then test the hypothesis and result.

In this context, spear phishing is an attack which is tailor-made, in that we know know their name, their email, something relatable such as an upcoming meeting, a comment on a post, their co-workers and deadline. The point is to make the person believe you and blindly trust the attacker as a genuine. There is no mass appeal, the attacker put time and effort into knowing their target. Context matters, the users and assets of identity involved in the attack

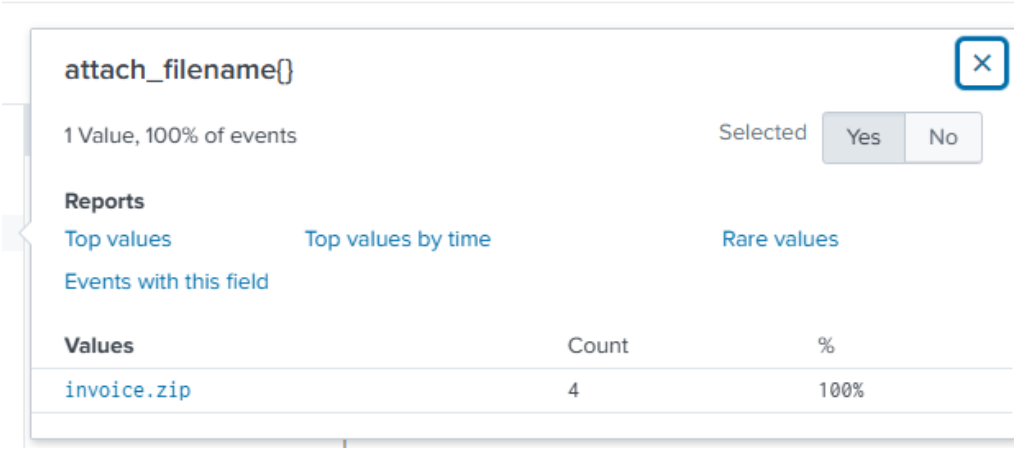
Attack vector

The delivery method being mostly email, our hypothesis questions include: Who the email was destined for, the time and subject and what was in the attachment.

#index=botsv2 sourcetype=stream:smtp date range 1st August 2017 till 31st August 2017

look at the attach_filename tab, we see a few files which are of interest.

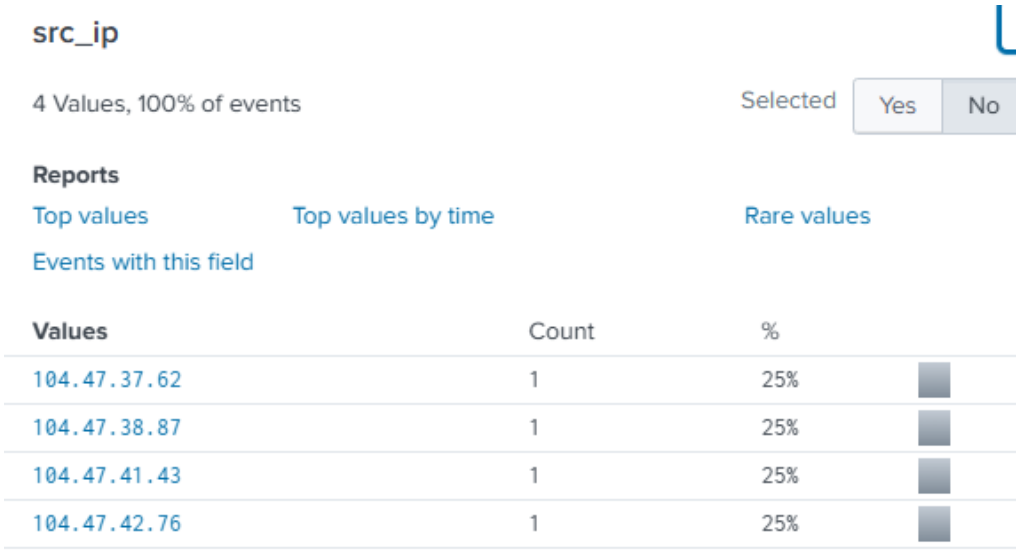
#index=botsv2 sourcetype=stream:smtp attach_filename{}=invoice.zip



Use of urgent language to encourage the user to download the attachment

```
<html>
<head>
<meta http-equiv=3D"Content-Type" content=3D"text/html; charset=3DUTF-8">
</head>
<body>
<div>
<div data-node-type=3D"line" id=3D"magicdomid2">
<div data-node-type=3D"line" id=3D"magicdomid2">As we have not received a =
service cessation letter, I am assuming that you might have accidentally =
overlooked this invoice &lsquo;02/160000506500 (Unpaid)&rsquo; for 10,000 =
GBP. Should you wish to bring an end to the agreement please let us know. =
Otherwise early withdrawal penalties will apply next month. </div>
<div data-node-type=3D"line" id=3D"magicdomid3"> </div>
<div data-node-type=3D"line" id=3D"magicdomid4">Pleaser refer to the =
attached document for payment details.</div>
</div>
</div>
```

The source IP of events associated with invoive.zip

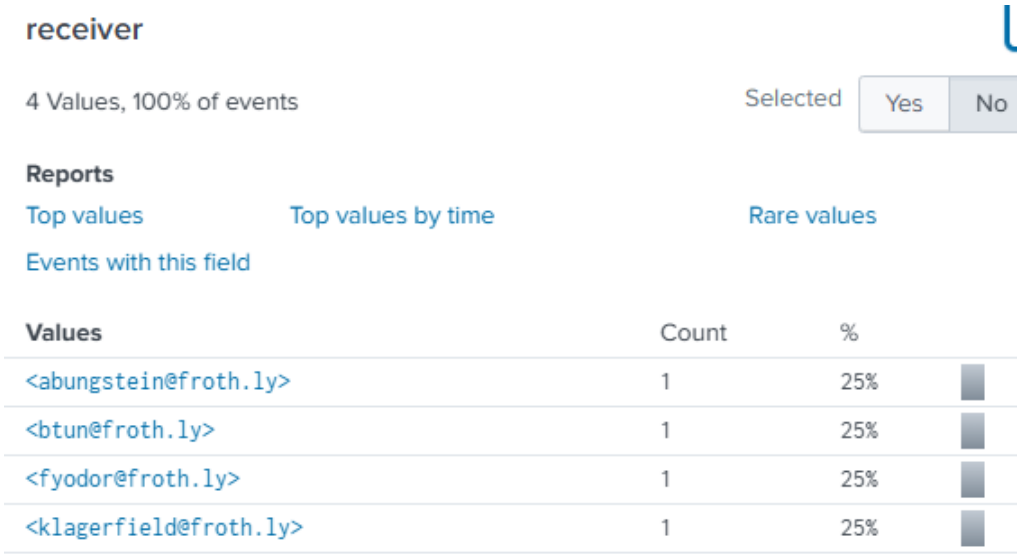


The content field gives us the process of sending and receiving mail.

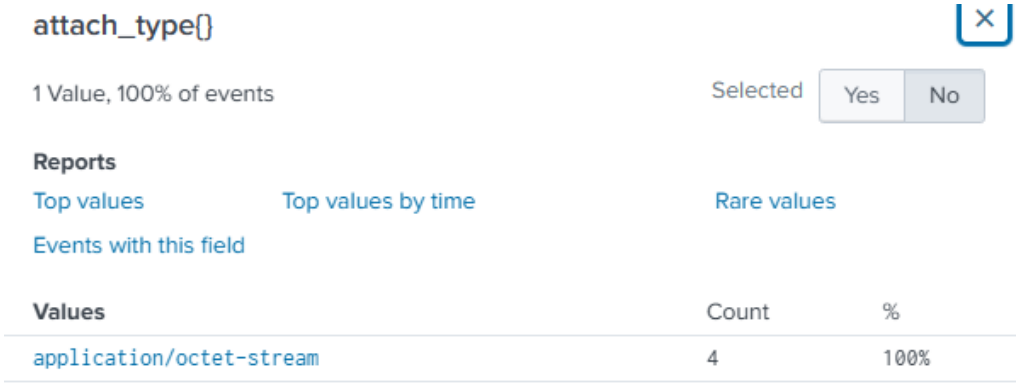
In the authentication headers, I retrieved the sender IP

```
 cipher=TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384_P256) id 15.1.1341.17 via
 Frontend Transport; Thu, 24 Aug 2017 03:27:28 +0000
Authentication-Results: spf=pass (sender IP is 185.83.51.21)
 smtp.mailfrom=smtp12.ymlpsvr.com; froth.ly; dkim=none (message not signed)
 header.d=none;froth.ly; dmarc=none action=none header.from=urinalysis.com;
Received-SPF: Pass (protection.outlook.com: domain of smtp12.ymlpsvr.com
 designates 185.83.51.21 as permitted sender) receiver=protection.outlook.com;
 client-ip=185.83.51.21; helo=smtp12.ymlpsvr.com;
```

The recipients of the email from the IP address 185.83.51.21



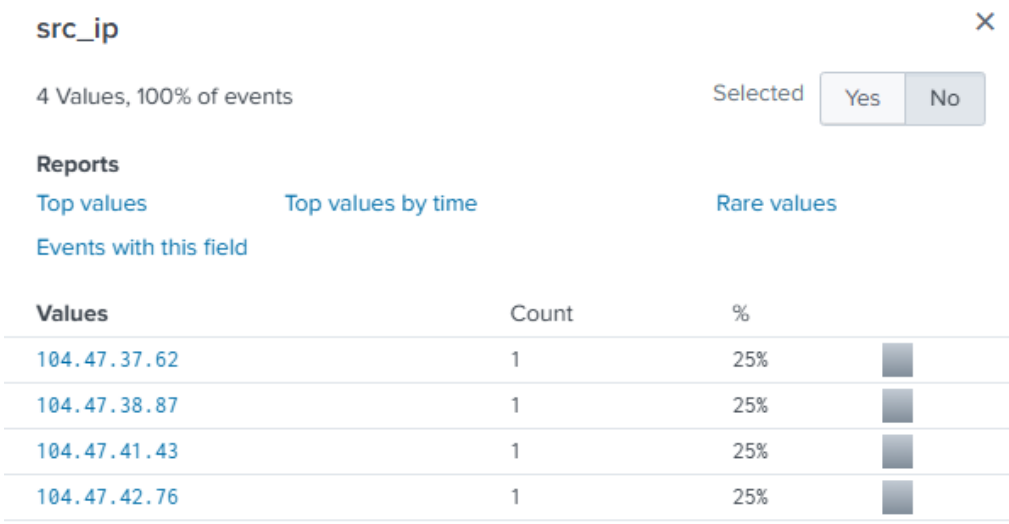
The application/octet-stream means the web browser could not classify the file thus naming it a generic binary file. This is often used in malware delivery to hide the real format such as .exe or.dll and the web browsers do not flag the file as malicious.



Let's co-relate if the recipients match the same source IP with a regex command

```
# index=botsv2 sourcetype=stream:smtp attach_filename{}=invoice.zip
| rex field=content "sender IP is (?<sender_ip>\d+.\d+.\d+.\d+)"
| search sender_ip=185.83.51.21
```

This search returns the above IP addresses



With open source intelligence, Whois return that the above are owned by Microsoft and geolocated in Redmond, America.

185.83.51.21 on AbuseIPDB has been reported for email spam and port scan.

ISP	YMLP BVBA
Usage Type	Commercial
ASN	AS201168
Hostname(s)	smtp12.ymlpsvr.com
Domain Name	ymlp.com
Country	Belgium
City	Brussels, Brussels Capital

The domain Urinalysis.com

DNS Records for urinalysis.com

Hostname	Type	TTL	Priority	Content
urinalysis.com	A	0		209.196.146.115
urinalysis.com	SOA	3600		ns1.epik.com support.epik.com 2019111101 10800 3600 604800 3600
urinalysis.com	NS	0		ns4.epik.com
urinalysis.com	NS	0		ns3.epik.com
www.urinalysis.com	A	0		209.196.146.115

MITRE ATT&CK⟶ Acquire infrastructure: web services ID T1583.006, specifically T1583

Hypothesis⟶ Has John smith sent any other emails within the month?

#index=botsv2 sourcetype=stream:smtp sender="Jim Smith <Jsmith@urinalysis.com>"
| table _time recipient subject attach_filename{} attach_size{} attach_content_decoded_md5_hash{}

Key Fields Where Sender is Jim Smith

_time	recipient	subject	attach_filename{}	attach_size{}	attach_content_decoded_md5_hash{}
2017-08-23 20:27:14.323	abungstein@froth.ly	Invoice	invoice.zip	22578	0fa0f1b660962d4a4d1cd6782a03db05
2017-08-23 20:27:29.837	btun@froth.ly	Invoice	invoice.zip	22578	0fa0f1b660962d4a4d1cd6782a03db05
2017-08-23 20:27:24.557	klagerfield@froth.ly	Invoice	invoice.zip	22578	0fa0f1b660962d4a4d1cd6782a03db05
2017-08-23 20:27:33.239	fyodor@froth.ly	Invoice	invoice.zip	22578	0fa0f1b660962d4a4d1cd6782a03db05
2017-08-10 20:25:03.369	abungstein@froth.ly	Invoice Doc	Malware Alert Text.txt	256	41099cf098c8e7655e9fd73b29e14d70
2017-08-10 20:24:51.513	fyodor@froth.ly	Invoice Doc	Malware Alert Text.txt	256	41099cf098c8e7655e9fd73b29e14d70
2017-08-10 20:24:45.500	klagerfield@froth.ly	Invoice Doc	Malware Alert Text.txt	256	41099cf098c8e7655e9fd73b29e14d70
2017-08-10 20:24:46.808	btun@froth.ly	Invoice Doc	Malware Alert Text.txt	256	ae87fac6adae1d69cec7e4284892c07a

More digging on the attachment malware.txt

index=botsv2 sourcetype=stream:smtp "attach_filename{}"="Malware Alert Text.txt" | table _time recipient subject content{}

```
src=3D"http://t.ymlp160.net/gwplhesyskwf/footer.gif" width=3D"1" =
border=3D"0">
</body></html>

--_32b7f1b0-8253-4f59-b2e8-a6ca6eaa5cb1_
Content-Type: application/octet-stream; name="Malware Alert Text.txt"
Content-Description: Malware Alert Text.txt
Content-Disposition: attachment; filename="Malware Alert Text.txt"
Content-Transfer-Encoding: base64

TWFsd2FyZSB3YXMgZGV0ZWN0ZWQgaW4gb25lIG9yIG1vcmUgYXR0YWNobWVudHMgaW5jbHVkZWQg
d2l0aCB0aGlzIGVtYWlsIG1lc3NhZ2UuIA0KQWN0aW9uOiBBbGwgYXR0YWNobWVudHMgaGF2ZSBi
ZWVuIHJlbW92ZWQuDQppbnZvaWNlLmRvYwkgVHJvamFuLlpWRUotMg0KaW52b2ljZS5kb2MJIE85
N00vRG9ub2ZmIXJmbg0K

--_32b7f1b0-8253-4f59-b2e8-a6ca6eaa5cb1_--
```

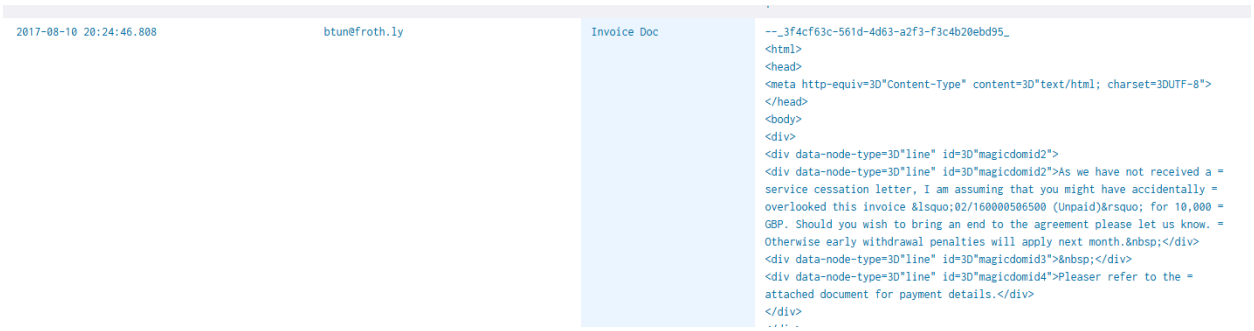
Base 64 decoded shows the attacker checked the possibility of .txt being flagged by an antivirus and removed therefore unsuccessful.

```
Malware was detected in one or more attachments included with this email message.
Action: All attachments have been removed.
invoice.doc  Trojan.ZVEJ-2
invoice.doc  O97M/Donoff!rfn
```

Comparing the emails

#index=botsv2 sourcetype=stream:smtp sender="Jim Smith jsmith@urinalysis.com"
| table _time recipient subject content_body{}
| sort recipient

With this sort, the body content is similar to the four emails with urgent language such as withdrawal penalties and refer to the attachment document. Invoice doc failed but invoice.zip was successful.

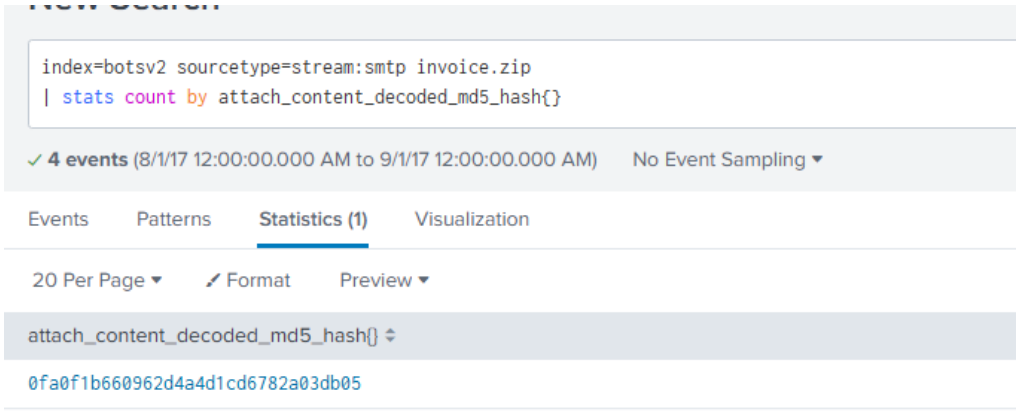


Attachment hash value shows 4 pdf.

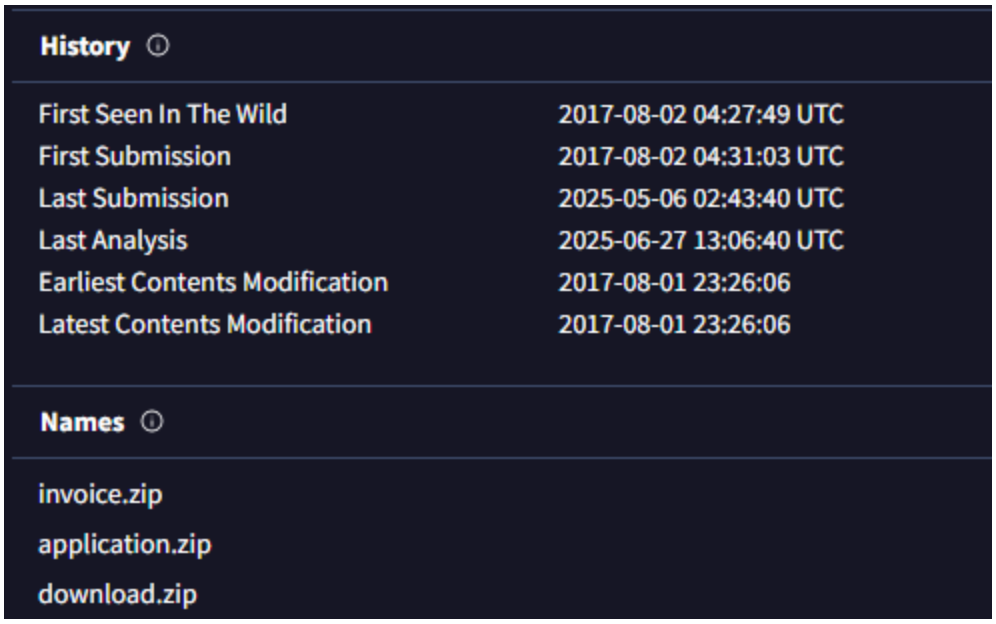
index=botsv2 sourcetype=stream:smtp invoice.zip

| stats count by attach_content_decoded_md5_hash{}

MD5 hash of all four recipients who have the same hash.



Checking using Virus total



The file (invoice.zip) was submitted early August and tested to ensure passing the antivirus before it became an attack vector.

Hunt for the execution of the malicious File execution. Hypothesis questions include:

what are the data sources ; what supporting info do we have ; what other indicators do we have;

what happened upon execution of the file.

More Data Sources

#index=botsv2 sourcetype!=stream:smtp invoice.zip

The host workstation being wrk-btun

In the sourcetype field, we get more logs to work with.

XmlWinEventLog:Microsoft-Windows-Sysmon/Operational	2	40%	
WinHostMon	1	20%	
WinRegistry	1	20%	
wineventlog			

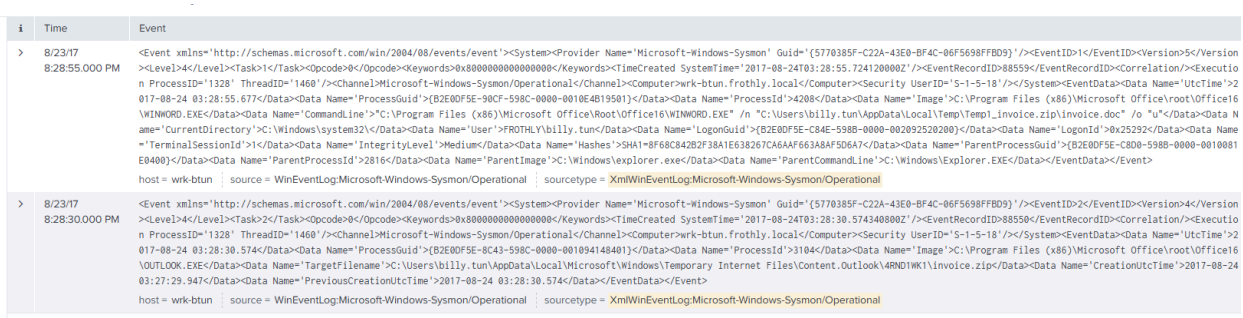
The registry shows the temp folder is where the malicious file was stored. This occurred on 08/23/20171 at 20:41:53 PM

#index=botsv2 sourcetype!=stream:smtp invoice.zip sourcetype=WinRegistry

```
08/23/2017 20:41:53.770
event_status="(0)The operation completed successfully."
pid=4208
process_image="c:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE"
registry_type="SetValue"
key_path="HKU\s-1-5-21-3348076501-352378380-2991248034-1115\software\microsoft\office\16.0\word\reading locations\document 0\file path"
data_type="REG_SZ"
data="C:\Users\billy.tun\AppData\Local\Temp\Temp1_invoice.zip\invoice.doc"
Collapse
host = wrk-btun | source = WinRegistry | sourcetype = WinRegistry
```

Event logs shows us a process creation occurred meaning execution of the invoice.pdf

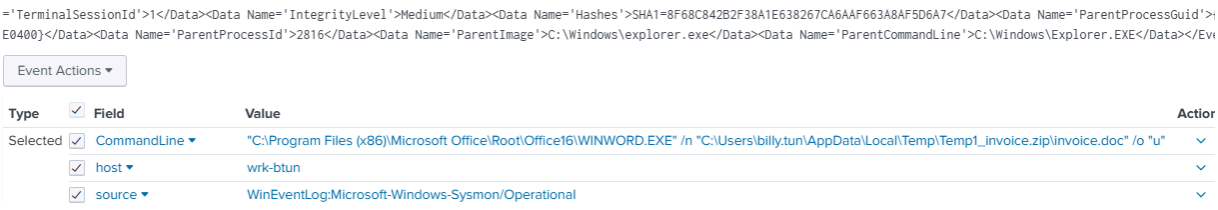
#index=botsv2 sourcetype!=stream:smtp invoice.zip sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational”



Windows Event logs

```
LogName=Security
SourceName=Microsoft Windows security auditing.
EventCode=4688
EventType=0
Type=Information
ComputerName=wrk-btun.frothly.local
TaskCategory=Process Creation
OpCode=Info
RecordNumber=57266
Keywords=Audit Success
Message=A new process has been created.

Subject:
        Security ID:            FROTHLY\billy.tun
        Account Name:           billy.tun
        Account Domain:         FROTHLY
        Logon ID:               0x25292

Process Information:
        New Process ID:         0x1070
        New Process Name:       C:\Program Files (x86)\Microsoft Office\root\Office16\WINWORD.EXE
        Token Elevation Type:   TokenElevationTypeLimited (3)
        Creator Process ID:     0xb00
        Process Command Line:   "C:\Program Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Users\billy.tun\AppData\Local\Temp\Temp1_invoice.zip\invoice.doc" /o "u"

Token Elevation Type indicates the type of token that was assigned to the new process in accordance with User Account Control policy.

Type 1 is a full token with no privileges removed or groups disabled.  A full token is only used if User Account Control is disabled or if the user is the built-in Administrator account or a serv
t.

Type 2 is an elevated token with no privileges removed or groups disabled.  An elevated token is used when User Account Control is enabled and the user chooses to start the program using Run as a
or.  An elevated token is also used when an application is configured to always require administrative privilege or to always require maximum privilege, and the user is a member of the Administra
p.

Type 3 is a limited token with administrative privileges removed and administrative groups disabled.  The limited token is used when User Account Control is enabled, the application does not requ
strative privilege, and the user does not choose to start the program using Run as administrator.
Collapse
```

Account_Domain = FROTHLY | Account_Name = billy.tun | ComputerName = wrk-btun.frothly.local | Security_ID = FROTHLY\billy.tun | host = wrk-btun | source = WinEventLog:Security | sourcetype = wineventlog
subject = A new process has been created



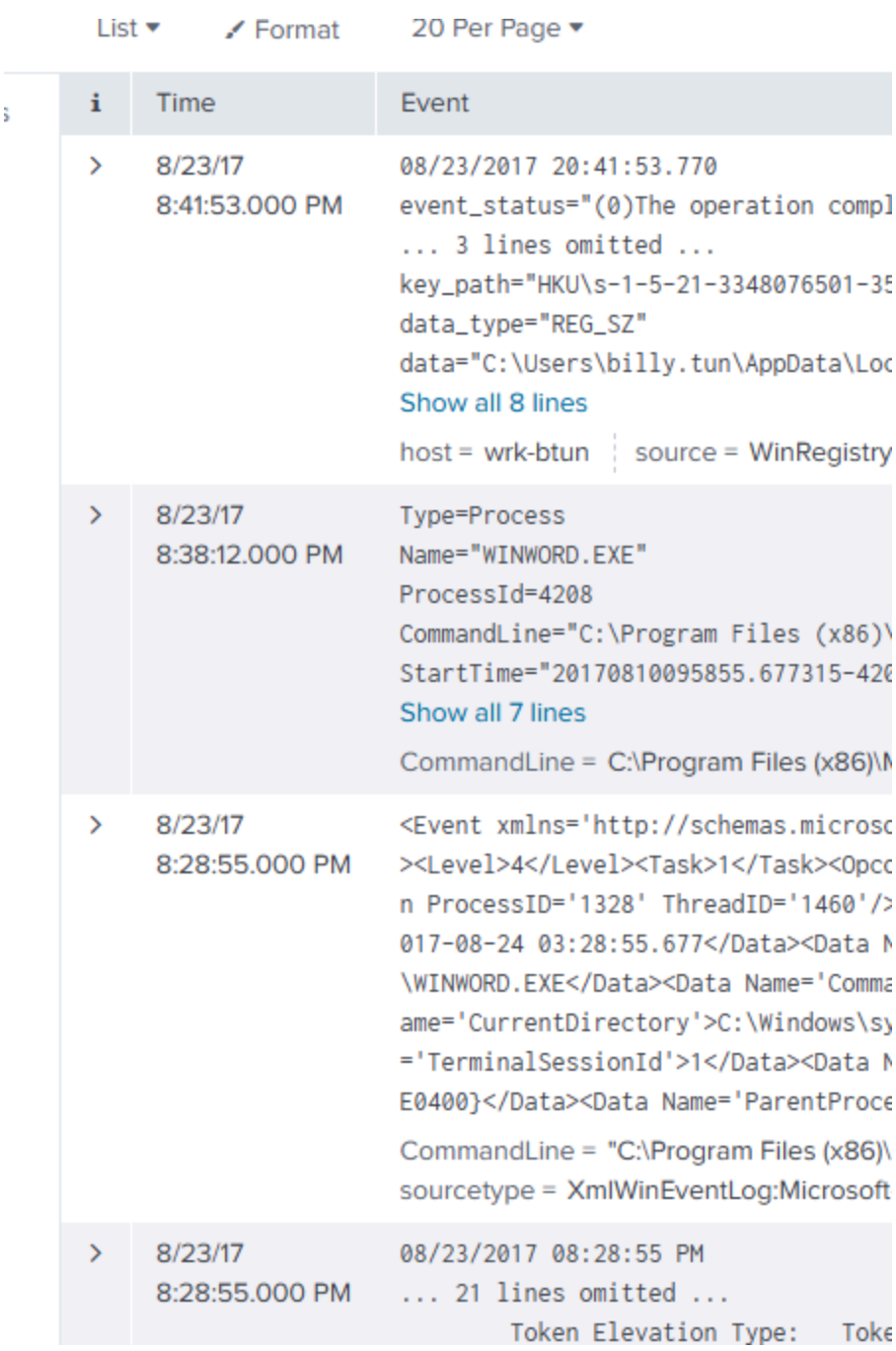
```
='TerminalSessionId'>1</Data><Data Name='IntegrityLevel'>Medium</Data><Data Name='Hashes'>SHA1=8F68C842B2F38A1E638267CA6AAF663A8AF5D6A7</Data><Data Name='ParentProcessGuid'>
E0400}</Data><Data Name='ParentProcessId'>2816</Data><Data Name='ParentImage'>C:\Windows\explorer.exe</Data><Data Name='ParentCommandLine'>C:\Windows\Explorer.EXE</Data></Ev
```

Event Actions

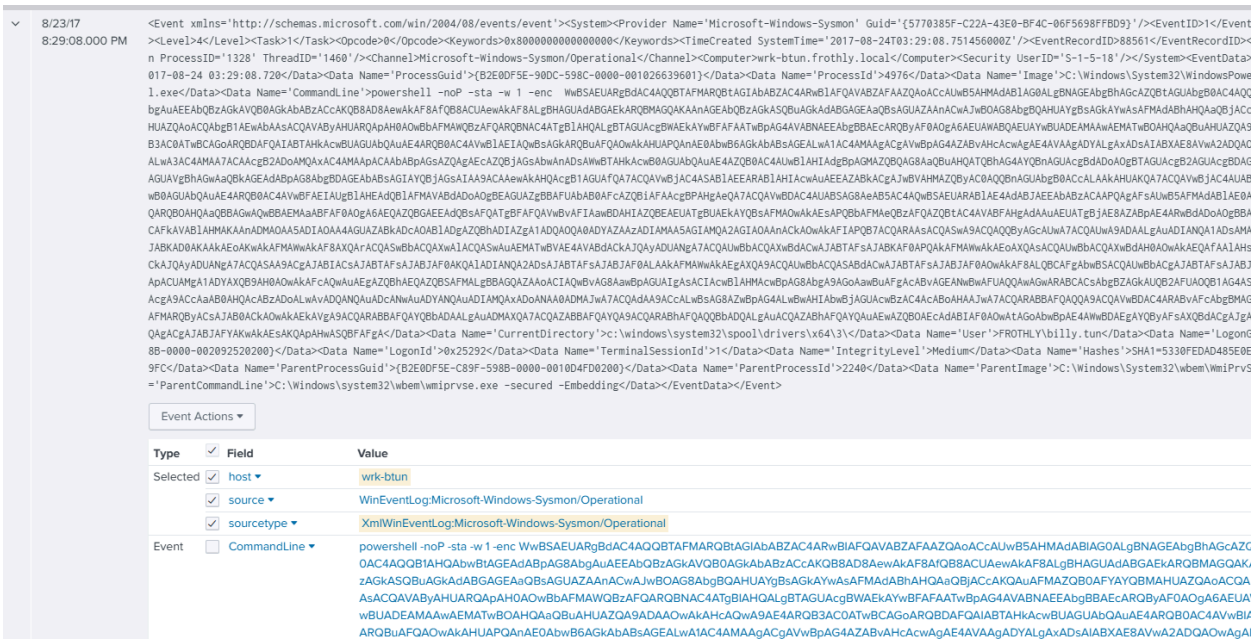
Type	Field	Value	Actior
Selected	CommandLine	"C:\Program Files (x86)\Microsoft Office\Root\Office16\WINWORD.EXE" /n "C:\Users\billy.tun\AppData\Local\Temp\Temp1_invoice.zip\invoice.doc" /o "u"	
	host	wrk-btun	
	source	WinEventLog:Microsoft-Windows-Sysmon/Operational	

The command execution is microsoft office winword.exe which opens the invoice.zip and inside is invoice.doc The lack of antivirus may mean there is a macros embedded or malicious payload.

Time Lapse→ Let's narrow the time lapse to when the first process was created and the activities after using sysmon time. The most activities happened between 8 PM to 9 PM on August 23 2017.



#index=botsv2 host=wrk-btun sourcetype="XmlWinEventLog:Microsoft-Windows-Sysmon/Operational" | reverse to see the events from last to first. Powershell encoded which is another Indicator of compromise, mapped to command and scripting .



The decoded text shows commands for credential harvesting.

Summary

The attacker:

1. Sent a spear phishing email with invoice.zip
2. User(Billy Tun) extracted and opened `invoice.doc`
3. Word process (`WINWORD.EXE`) executed the file from the Temp directory

4. Registry logs captured the document path
5. Sysmon logs showed **PowerShell scripts**, suggesting credential harvesting activity

Reference

https://www.splunk.com/en_us/blog/learn/spear-phishing.html

https://www.sans.org/white-papers/who-what-where-when-why-how-effective-threat-hunting/

https://www.techradar.com/pro/security/chinese-hackers-hit-taiwan-semiconductor-manufacturing-in-spear-phishing-campaign?utm_source=chatgpt.com

https://www.youtube.com/watch?v=oCkgJlxYujs