How could somebody on the defense side (i.e., in charge of the network security) make the most of whois against attackers?

- that you are exposing too much information about yourself in your own domain info (owners name, address, phone number, etc.)
- how long domains have been registered so that you can use automated tools to block domains that are very young (and likely to be malicious if suddenly appearing in emails)
- who to contact in the event that a legitimate domain is sending spam or hosting malicious content
- To complying with ICANN's policies, there must be *some* contact information in the WHOIS record for your domain, but it doesn't have to be your personal data.

# 1.2 - Netcraft - go to resources -> Site report

# 1.2 Host command

host tieto.com

tieto.com has address 217.114.85.70 tieto.com mail is handled by 10 ebb07.tieto.com. tieto.com mail is handled by 10 ebb09.tieto.com. tieto.com mail is handled by 10 ebb08.tieto.com. tieto.com mail is handled by 10 ebb10.tieto.com.

Alternative:

- dig:
  - dig tito.com MX
  - ; <<>> DiG 9.16.37-Debian <<>> tieto.com MX
  - ;; global options: +cmd
  - ;; Got answer:
  - ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41838
  - ;; flags: qr rd ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0
  - ;; WARNING: recursion requested but not available

# ;; QUESTION SECTION:

# ;; ANSWER SECTION:

tieto.com.	0	IN	MX	10 ebb07.tieto.com.
tieto.com.	0	IN	MX	10 ebb09.tieto.com.
tieto.com.	0	IN	MX	10 ebb08.tieto.com.
tieto.com.	0	IN	MX	10 ebb10.tieto.com.

- ;; Query time: 0 msec
- ;; SERVER: 172.31.96.1#53(172.31.96.1)

- ;; WHEN: Sat May 13 11:39:39 CEST 2023
- ;; MSG SIZE rcvd: 160
  - dig tieto.com MX +short
    - 10 ebb07.tieto.com.
    - 10 ebb09.tieto.com.
    - 10 ebb08.tieto.com.
    - 10 ebb10.tieto.com.
- nslookup

nslookup

> set type=mx

> tieto.com

Server: 10.0.0.138 Address: 10.0.0.138#53

Non-authoritative answer:

tieto.commail exchanger = 10 ebb07.tieto.com.tieto.commail exchanger = 10 ebb09.tieto.com.tieto.commail exchanger = 10 ebb08.tieto.com.tieto.commail exchanger = 10 ebb10.tieto.com.

## Are mail servers hosted by the same company? Depending on the company, the answer to this question can be "yes" or "no". Considering each of these possibilities, does it make sense targeting mail servers as potential vectors for penetration attacks?

In case that the mail server is hosted by the same company the answer is clearly yes as this might give access to other servers as well if the different company servers are not properly separated / isolated from each other. While additionally giving additional information via email.

If the mail server is hosted by a different company the answer is still yes as this can open up the possibility for supply chain attacks or give more information about the company. However, in this case it is really important to get a permit to attack for this part as well because otherwise this is not legal.

Why email: password resets via email, sensitive data, company secrets, address books, ...

## 2.2 Port probing



10.0.0.1 -> 10.0.0.2 445 - Destination unreachable ICMP 10.0.0.1 -> 10.0.0.2 113 - Paket dropped no response = retransmission 10.0.0.1 -> 10.0.0.2 9920 - RST (destination port is closed) Imagine using Wireshark for checking all the traffic passing through an intermediate routing device. Do you think that you could detect hosts performing horizontal scanning? And vertical scanning? Do you consider Wireshark as a suitable tool for analyzing large amounts of network traffic data? Why?

The problem with wireshark is that it is really good for static analysis but not so much for dynamic analysis. For dynamic analysis the new packets are constantly coming in which makes analysing the data really hard. For this purpose tools like suricata or snort are better suited especially because they also have alerting functionalities that make it easier to focus only on the important packets.

However, for static data wireshark it is possible to detected horizontal and vertical scanning as one can filter accoring to some tcp port or soucre ip and also specify ip ranges. But this comes with a lot of effort (think about legitimate connections). The problem is you cannot specify complex rules like: if from one source ip there are requests to greater x ips alert me.

3.1 - Horizontal scan addresses are varied

				· _ · · · ·		
📕 ar	<b>)</b>					
No.	Time	Source	Destination	Protocol Lene	gth Info	
	1 0.000000	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	3 192.168.1.0? Tell 192.168.0.72
	2 0.000041	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	3 192.168.2.0? Tell 192.168.0.72
	3 0.000052	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	5 192.168.3.0? Tell 192.168.0.72
	4 0.000063	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	5 192.168.4.0? Tell 192.168.0.72
	5 0.000072	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	: 192.168.5.0? Tell 192.168.0.72
	6 0.000083	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	: 192.168.6.0? Tell 192.168.0.72
	7 0.000094	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	3 192.168.7.0? Tell 192.168.0.72
	8 0.000104	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	3 192.168.8.0? Tell 192.168.0.72
	9 0.000114	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	5 192.168.9.0? Tell 192.168.0.72
	10 0.000124	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	3 192.168.10.0? Tell 192.168.0.72
	11 0.000134	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	3 192.168.11.0? Tell 192.168.0.72
	12 0.000144	fa:95:9c:8e:2d:34	Broadcast	ARP	42 Who has	3 192.168.12.07 Tell 192.168.0.72
	40.0.00455	£05.000d.0d	Durandarat	400	AO Miles less	400 400 40 00 T-11 400 400 0 70

The size of an ARP request or reply packet is 28 bytes **and more importantly not really a calculation is needed when receiving the response**. (DOS attack) <u>https://blog.radware.com/security/2012/02/ddos-attacks-myths/</u>

_										
	I http									
No.	Time	Source	Destination	Protocol	Length	Info				
	131 0.093404	192.168.54.7	192.168.0.72	HTTP	103	2 HTTP/1.0 200 OK	(text/html)			
	133 0.094132	192.168.54.7	192.168.0.72	HTTP	103	2 HTTP/1.0 200 OK	(text/html)			
	141 0.097611	192.168.54.7	192.168.0.72	HTTP	103	2 HTTP/1.0 200 OK	(text/html)			
	143 0.097710	192.168.54.7	192.168.0.72	HTTP	103	2 HTTP/1.0 200 OK	(text/html)			
	145 0.098082	192.168.54.7	192.168.0.72	HTTP	103	2 HTTP/1.0 200 OK	(text/html)			
	153 0.399239	192.168.0.72	192.168.54.7	HTTP	25	6 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	157 0.421471	192.168.54.7	192.168.0.72	HTTP	115	1 HTTP/1.0 200 OK	(text/html)			
	198 0.499588	192.168.0.72	192.168.54.7	HTTP	25	9 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	200 0.499606	192.168.0.72	192.168.54.7	HTTP	25	6 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	201 0.499622	192.168.0.72	192.168.54.7	HTTP	25	5 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
+	204 0.499717		192.168.54.7	HTTP		5 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	208 0.500936	192.168.0.72	192.168.54.7	HTTP	25	5 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	212 0.501496	192.168.0.72	192.168.54.7	HTTP	25	5 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	216 0.504135	192.168.0.72	192.168.54.7	HTTP	25	5 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	220 0.505666	192.168.0.72	192.168.54.7	HTTP	25	5 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	224 0.508498	192.168.0.72	192.168.54.7	HTTP	25	4 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	228 0.510051	192.168.0.72	192.168.54.7	HTTP	25	4 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	232 0.513032	192.168.0.72	192.168.54.7	HTTP	25	3 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	236 0.514002	192.168.0.72	192.168.54.7	HTTP	25	6 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
	240 0.517350	192.168.0.72	192.168.54.7	HTTP	25	4 POST / HTTP/1.0	(application/x-www-form-urlencoded)			
→ F	rame 204: 255 byte	s on wire (2040 bit	s), 255 bytes captured	(2040 bit	s)					
- → E	▶ Ethernet II, Src: fa:95:9c:8e:2d:34 (fa:95:9c:8e:2d:34), Dst: 9e:36:56:fd:3f:8c (9e:36:56:fd:3f:8c)									
▶ Internet Protocol Version 4, Src: 192.168.0.72, Dst: 192.168.54.7										
Transmission Control Protocol, Src Port: 43372, Dst Port: 80, Seq: 1, Ack: 1, Len: 189										
- > F	Hypertext Transfer Protocol									
- F	HTML Form URL Encoded: application/x-www-form-urlencoded									
	Form item: "username" = "Celina_Shelby"									
	Form item: "password" = "Hunter"									

## 3.1 - Bruteforce

Find butofocre creds: http && (http.content\_length > 1000 || http.content\_length < 800) !data-text-lines contains "Invalid Credentials" && http.response.code == 200 && !data-text-lines contains "Login"

1	224 0.508498	192.168.0.72	192.168.54.7	HTTP	254 P0ST	/ HTTP/1.0	(application/x-www-	form-urlenco	ded)		
÷	228 0.510051	192.168.0.72	192.168.54.7	HTTP	254 POST	/ HTTP/1.0	(application/x-www-	form-urlenco	ded)		
	232 0.513032	192.108.0.72	192.108.54.7	нтр	253 PUST	/ HITP/1.0	(application/x-www-	form-urlenco	ded)		
	240 0.517350	192.168.0.72	192.168.54.7	HTTP	254 P0ST	/ HTTP/1.0	(application/x-www-	form-urlenco	ded)		
×	Frame 228: 254 byt	es on wire (2032 b	its), 254 bytes captur	ed (2032 bits)							
÷	Ethernet II, Src: fa:95:9c:8e:2d:34 (fa:95:9c:8e:2d:34), Dst: 9e:36:56:fd:3f:8c (9e:36:56:fd:3f:8c)										
Ľ	Internet Protocol Version 4, Src: 192.168.0./2, Dst: 192.168.04./										
	Hypertext Transfer	Protocol	DIC. 43304, DSC FOIL.	50, Seq. 1, AC	K. 1, Len. 1	100					
*	HTML Form URL Enco	HTML Form URL Encoded: application/x-www-form-urlencoded									
	Form item: "user	name" = "Celina_Sh	elby"								
	▶ Form item: "pass	word" = "Knorr"									
	0		Wireshark · Follow	HTTP Stream (to	p.stream eq 17	′879)•pcap3_t	eam15.pcap			000	
1											
2	POST / HTTP/1.0	7									
1	HOSL: 192.108.54	./ ]]a/5 0 (Hydra)									
	Content-Length:	35									
-	Content-Type: ap	plication/x-www-f	orm-urlencoded								
7	Cookie:										
	userse of the state	Deth@sees.comd_D									r=
7	Username=Charla_	kotn&password=kec xt/html:_charset=	10HTTP/1.0 200 OK								
7	Content-Length:	1701	uti-o								
•	Server: Werkzeug	/0.14.1 Python/2.	7.16								
•	Date: Mon, 04 Ma	y 2020 18:36:32 G	MT								se
1	1.1.2.										am
7	<ntml></ntml>										9
	<title>Googa</title>	l - main name <td>tle&gt;</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	tle>								
7	<meta content<="" name="&lt;/td&gt;&lt;td&gt;viewport" td=""/> <td>="width=device-width,</td> <td>initial-sca</td> <td>le=1.0"&gt;</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	="width=device-width,	initial-sca	le=1.0">							
1	<link href="&lt;/td&gt;&lt;td&gt;https://maxcdn.boots&lt;/td&gt;&lt;td&gt;trapcdn.com/b&lt;/td&gt;&lt;td&gt;ootstrap/3.&lt;/td&gt;&lt;td&gt;4.0/css/boo&lt;/td&gt;&lt;td&gt;tstrap.min.css" rel="s&lt;/td&gt;&lt;td&gt;tylesheet"/>										
•	10 h										
	<script_src=< td=""><td>"https://ajax.goo</td><td><pre>gleanis.com/aiax/lib</pre></td><td>s/iquery/3.3.</td><td>l/iquery.mi</td><td>n.is"&gt;<td>int&gt;</td><td></td><td></td><td></td><td></td></td></script_src=<>	"https://ajax.goo	<pre>gleanis.com/aiax/lib</pre>	s/iquery/3.3.	l/iquery.mi	n.is"> <td>int&gt;</td> <td></td> <td></td> <td></td> <td></td>	int>				
	<script src="&lt;/td"></script>										

2 -

wireshark packet 10

PING nutzt das Protokoll ICMP, welches auf der Ebene 3 im OSI-Schichtenmodell angesiedelt ist, wie IP. Es ist also etwas "tiefer" als TCP und UDP, die auf Ebene 4 sind. ICMP kennt daher auch keine "Ports".

✓ Frame 10: 60 bytes on wire (480 bits), 42 bytes captured (336 bits) on interface unknown, id 0	
Section number: 1	
> Interface id: 0 (unknown)	
Encapsulation type: Ethernet (1)	
Arrival Time: Jan 1, 2019 06:03:30.014402000 W. Europe Standard Time	
[Time shift for this packet: 0.000000000 seconds]	
Epoch Time: 1546319010.014402000 seconds	
[Time delta from previous captured frame: 0.000014000 seconds]	
[Time delta from previous displayed frame: 0.000014000 seconds]	
[Time since reference or first frame: 0.000057000 seconds]	
Frame Number: 10	
Frame Length: 60 bytes (480 bits)	
Capture Length: 42 bytes (336 bits)	
[Frame is marked: False]	
[Frame is ignored: False]	
[Protocols in frame: eth:ethertype:ip:icmp]	
[Coloring Rule Name: ICMP]	
[Coloring Rule String: icmp    icmpv6]	
> Ethernet II, Src: JuniperN_7a:66:f0 (88:e0:f3:7a:66:f0), Dst: Cisco_19:6a:52 (64:f6:9d:19:6a:52)	
Internet Protocol Version 4, Src: 203.74.52.109, Dst: 202.153.212.143	
0100 = Version: 4	
0101 = Header Length: 20 bytes (5)	
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	
Total Length: 32	
Identification: 0xc954 (51540)	
> 010 = Flags: 0x2, Don't fragment	
0 0000 0000 0000 = Fragment Offset: 0	
Time to Live: 59	
Protocol: ICMP (1)	
Header Checksum: 0xd7a7 [validation disabled]	
[Header checksum status: Unverified]	
Source Address: 203.74.52.109	
Destination Address: 202.153.212.143	
> Internet Control Message Protocol	

#### Install Go-Flows

- 1) Clone repo
- 2) go build
- 3) go install
- 4) locate go-flows binary (was in ~/go/bin)
- 5) execute command: ./go-flows run features

/mnt/c/Users/alexh/Downloads/pcap2pkts.json export csv Ex2\_team15.csv source libpcap /mnt/c/Users/alexh/Desktop/Ex2\_team15.pcap

Remember that here we have extracted flows within a time-frame of 10 seconds. Can you think about legitimate and ilegitimate situations for case (c), i.e., a source sending traffic to many different destinations in a short time?

- Legitimate
  - Server (sends to many clients)
- Illegitimate
  - Horizontal Scan
  - Botnet control traffic

You can additionally count the number of flows that show TCP, UDP, ICMP, and other IP protocols as "mode" protocol. Do you think that you will get a similar proportion as in [rep-11]? Beyond answering "yes" or "no", think about reasons that might make such proportions similar or different (there are some that are worth considering). mode: value that appears the most mode count: how often is the mode value present

rep23: diagram does not tell us anything.... the number in the diagram works however (even though it is off by approx. 10 or so) 176977 works with 176964

rep23: solution - use filter

part 2: srcip address: 93.91.224.215 total # of sent packets: 35397