INCIDENT REPORT

Cyberlabs Final Exam

index = inc06

Prepared by

David Belayev Gal

August 2026

Table of Contents

Executive Summary

A PowerShell script (`ransomware.ps1`) was executed on host `WS01.corp.local` under the `CORP\cyberlabs` account on August 3rd 2026, generating over 5,700 cmdlet execution events consistent with disk-encryption activity. Log analysis reconstructs a multi-stage compromise involving four accounts on the same host:

1. **johnsmith** — earliest observed malicious activity (July 16th), running discovery commands from a local session.
2. **Administrator** — used less than two hours later to weaken PowerShell execution restrictions and query the Sysmon process, consistent with defense evasion.
3. **itadmin** — used to create the `cyberlabs` account, disable Sysmon, delete files, and open the firewall for RDP; later used to execute a payload flagged by Windows Defender as a Meterpreter trojan.
4. **cyberlabs** — the account under which `ransomware.ps1` was ultimately executed.

The initial access vector and the identity behind source IP `10.0.2.7` remain unresolved.

Timeline

Time	Event
2026-07-16 11:48	First event recorded under the WS01.corp.local hostname, consistent with the point at which the host joined the domain (Figure 10)
2026-07-16 11:49	johnsmith local logon on WS01.corp.local (Figure 18)
2026-07-16 12:09	johnsmith executes script with discovery commands from Downloads folder (Figures 16, 17)
2026-07-16 13:50	Administrator executes commands weakening PowerShell execution restrictions and querying the Sysmon process (Figure 19)
2026-07-31 13:39	itadmin executes script performing directory traversal, file deletion, stopping Sysmon, and enabling an RDP firewall rule (Figure 20)
2026-07-31 13:45	Successful RDP connection to itadmin from source IP 10.0.2.7 (Figure 6)
2026-08-03 14:05	itadmin executes payload.exe from AppData/Temp, quarantined by Windows Defender as a Meterpreter trojan; occurred less than an hour before the cyberlabs session (Figure 22)
2026-08-03 ~14:47:09	itadmin executes new-domain-user.ps1, creating the cyberlabs account, seconds before the RDP session to cyberlabs begins (Figure 15)
2026-08-03 14:47:33	RDP connection and logon to cyberlabs from source IP 10.0.2.7; start of cyberlabs session (Fgures 6, 7, 8)
2026-08-03 14:47:33 – 14:51	cyberlabs session active; ransomware.ps1 executed, generating thousands of cmdlet execution events (800, 4103) alongside BitLocker module invocation (Figures 8, 9)
2026-08-03 14:51	RDP logoff for cyberlabs; end of session (Figure 8)

Indicators of Compromise

Type	Name	Meaning
File	`C:\Users\cyberlabs\Downloads\ransomware.ps1`	Malicious PowerShell script executed under cyberlabs; associated with BitLocker module invocation, consistent with disk-encryption intent
File	`new-domain-user.ps1`	Script executed by itadmin that created the cyberlabs account, immediately preceding the cyberlabs session (Figure 15)
File	`C:\Windows\Temp\payload.exe`	Executable quarantined by Windows Defender, detected as a Meterpreter trojan; executed by itadmin (Figure 22)
File	`C:\Users\johnsmith\Desktop\Clear-Windows-LabLogs.ps1`	Log-clearing script, consistent with anti-forensic/defense-evasion activity
File	`C:\Users\johnsmith\Downloads\Generate-NormalActivity.ps1`	Script observed in the environment; purpose not confirmed from available logs — suspected obfuscation. Flagged for follow-up
File	`Z:\coll_scripts\log_collection.ps1`	Script on the Z: partition executed by itadmin; contents and purpose not fully reviewed — flagged for follow-up
Account	`CORP\cyberlabs`	Account created via new-domain-user.ps1; used to execute ransomware.ps1
Account	`CORP\itadmin`	Used to create the cyberlabs account, disable Sysmon, delete files, open RDP access, and execute the Meterpreter payload
Account	`CORP\Administrator`	Used to weaken PowerShell execution restrictions and query the Sysmon process
Account	`CORP\johnsmith`	Earliest observed account in the compromise chain; used to run discovery commands
Host	`WS01.corp.local`	Sole host on which all identified malicious script execution occurred
Network	`10.0.2.7`	Source IP for RDP connections to both cyberlabs and itadmin; identity unresolved in available logs

Type	Name	Meaning
Detection	`Windows Defender — Meterpreter trojan`	Malicious code detection tied to payload execution by itadmin (Figure 21, 22)
Event ID	`4103`	PowerShell module logging — records cmdlet invocation with parameters; a primary signal used throughout to trace command execution
Event ID	`800`	PowerShell pipeline execution completed; paired with 4103 as the two dominant event types powershell traffic
Event ID	`1149`	RDP network-level authentication success; used to identify source IP 10.0.2.7
Event ID	`1116 / 1117`	Windows Defender malware detection / remediation action; flags the Meterpreter payload execution by itadmin
Command line	`powershell.exe -Command {(Get-ExecutionPolicy) -ne 'AllSigned'} {Set-ExecutionPolicy -Scope Process Bypass}; & 'ransomware.ps1'`	Execution policy bypass pattern used to launch ransomware.ps1 (Figure 9 payload field)
Command line	`Set-MpPreference -DisableRealtimeMonitoring $true`	Windows Defender real-time protection disabled prior to payload execution (Figure 15)
Command line	`Enable-NetFirewallRule -DisplayGroup 'Remote Desktop'`	Firewall modification opening RDP access, executed by itadmin (Figure 20)
Command line	`Set-StrictMode -Version 1 -Off`	PowerShell guardrail/strict-mode weakening, executed by multiple users (Figure 19)
Command line	`sc query sysmon, sc query sysmon64`	Sysmon service enumeration — reconnaissance against the host's own defensive tooling (Figure 19)

Initial Alert

The alert `index=inc06 "ransomware"` shows a PowerShell script file called `ransomware.ps1` that triggered 5686 Windows PowerShell event ID 4103 and 800 events on the `WS01.corp.local` host, running as `CORP\cyberlabs`.

```
index=inc06
| rename System.EventID as event
| rename System.Computer as computer
| rename EventData.* as *
| search ransomware.ps1
| stats count by event
```

event	count
400	1
403	1
4100	12
4103	2543
600	7
800	3143

Figure 1: Event ID frequency in 'ransomware.ps1' traffic.

```
index=inc06
| rename System.EventID as event
| rename System.Computer as computer
| rename EventData.* as *
| search CORP cyberlabs
| stats count by event
| sort - count
```

event	count
800	3143
4103	2543
4100	12
600	7
1149	1
21	1
22	1
24	1
400	1
403	1
41	1
42	1

Figure 2: Identical event ID frequency in the cyberlabs account as in the ransomware.ps1 traffic, indicating that the ransomware.ps1 traffic is associated with the cyberlabs user account.

```
index=inc06 cyberlabs
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
| dedup computer
| table computer
```

computer
WS01.corp.local

Figure 3: All cyberlabs activity originated from WS01.corp.local, indicating that the cyberlabs user activity corresponds to execution of ransomware.ps1 on WS01.corp.local.

Account Enumeration

Enumerating all existing user accounts in the environment is necessary to ensure the completeness of the investigation.

The query below excludes `S-1-5-18` (Local System) and `S-1-5-20` (Network Service) — these are well-known Windows SIDs for built-in service accounts, not domain users. Excluding them removes non-attributable service noise and isolates activity performed under actual user credentials, without discarding any adversary-relevant events – no malicious activity was spotted from these accounts.

```
index=inc06
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
         System.Security_attributes.UserID AS userid
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| search computer=WS01.corp.local AND NOT userid IN (S-1-5-18, S-1-5-20)
| stats values(userid) by event
```

```
800    CORP\Administrator
       CORP\SYSTEM
       CORP\cyberlabs
       CORP\itadmin
       CORP\johnsmith
```

```
4103   S-1-5-21-4260785253-2099098590-1992057404-1105
       S-1-5-21-4260785253-2099098590-1992057404-1126
```

Figure 4: A comprehensive list of users who executed commands. RID 1105 resolves to itadmin and RID 1126 resolves to cyberlabs; this mapping was confirmed by cross-referencing the SID-to-username resolution against the account's observed logon events (e.g., Figures 6–8), though the underlying resolution query was not captured in a screenshot.

```
index=inc06 latest="07/16/2026:11:48:07"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
| search computer=WS01 event=21
| sort _time
| table _time address user
```

_time	address	user
2026-07-15 15:32:09.517	LOCAL	DESKTOP-ET7845K\User01
2026-07-15 15:25:04.415	LOCAL	DESKTOP-ET7845K\defaultuser0

Figure 5: More users from another computer — not domain related. They follow the same pattern shown here and don't appear involved in the attack.

Therefore, the relevant user accounts are:

- CORP\cyberlabs
- CORP\itadmin
- CORP\Administrator
- CORP\johnsmith

Attack Source

Figure 2 shows a single 1149 event — a successful RDP network-level handshake. Investigating all 1149 events shows two connections from the same source IP, 10.0.2.7 — one to cyberlabs and one to itadmin, supporting the involvement of both accounts in the same attack.

```
index=inc06
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Param1 AS username
         UserData.EventXML.Param3 AS sourceip
| search event=1149
| table _time computer username sourceip
```

_time	computer	username	sourceip
2026-08-03 14:47:33.958	WS01.corp.local	cyberlabs	10.0.2.7
2026-07-31 13:45:55.659	WS01.corp.local	itadmin@corp.local	10.0.2.7

Figure 6: RDP connection attempt to cyberlabs and itadmin from the same IP address.

The IP 10.0.2.7 is not mentioned in any other logs. At this stage of the investigation the source of the RDP connections was a reasonable early lead to pursue; it did not resolve further within the available log sources and is carried forward as an open item (see Follow-up section) rather than pursued further here.

Cyberlabs Session

The logs contain evidence of a single RDP session to cyberlabs, starting on August 3rd 2026, between 14:47:33 and 14:51:02.

```
index=inc06
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Param1 AS username
         UserData.EventXML.Param3 AS sourceip
| search cyberlabs
| stats min(_time) as mintime, max(_time) as maxtime, count
| convert ctime(mintime) ctime(maxtime)
```

mintime	maxtime	count
08/03/2026 14:47:33.958123	08/03/2026 14:51:02.077671	5713

Figure 7: Start and finish time of cyberlabs logs. This timeframe alone does not establish a single session or that it was completed — see Figure 8.

```
index=inc06 earliest="08/03/2026:14:47:33"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Param1 AS username
         UserData.EventXML.Param3 AS sourceip
| search cyberlabs
| sort _time
| stats min(_time) as mintime, max(_time) as maxtime, count by event
| convert ctime(mintime), ctime(maxtime)
```

event	mintime	maxtime	count
1149	08/03/2026 14:47:33.958123	08/03/2026 14:47:33.958123	1
21	08/03/2026 14:47:51.160637	08/03/2026 14:47:51.160637	1
22	08/03/2026 14:47:51.365843	08/03/2026 14:47:51.365843	1
24	08/03/2026 14:51:02.077671	08/03/2026 14:51:02.077671	1
400	08/03/2026 14:50:44.758720	08/03/2026 14:50:44.758720	1
403	08/03/2026 14:50:52.096840	08/03/2026 14:50:52.096840	1
41	08/03/2026 14:47:35.735252	08/03/2026 14:47:35.735252	1
4100	08/03/2026 14:50:50.734784	08/03/2026 14:50:50.884741	12
4103	08/03/2026 14:50:46.932373	08/03/2026 14:50:51.962428	2543
42	08/03/2026 14:47:48.703236	08/03/2026 14:47:48.703236	1
600	08/03/2026 14:50:44.754084	08/03/2026 14:50:45.769618	7
800	08/03/2026 14:50:45.240924	08/03/2026 14:50:51.955946	3143

Figure 8: Specified last figure's query to show number of logs per event id. A single successful RDP logon (21) event matches the start time of the cyberlabs logs, and a single successful RDP logoff (24) matches the end time, indicating one complete session from logon to logoff.

Figure 8 shows that `ransomware.ps1` generated thousands of cmdlet execution attempt events (800, 4103). The high ratio of total execution attempts to failure events (403, 4100) suggests the execution was largely successful. Figure 14 also shows a BitLocker script invoked by the cyberlabs account (and others), consistent with disk-encryption intent.

Confidence: Medium. The 800 events are intertwined with 4100 error events in the same timeframe, so it cannot be confirmed that the script executed exactly as intended — see Figure 9.

```
index=inc06
| rename System.EventID as event
| rename System.Computer as computer
| rename EventData.* as *
| search ransomware.ps1 event=4100
```

Figure 9: Every 4100 event shares the same payload field, so showing only details of one is enough. The script name is ransomware.ps1 and the payload field says "The running command stopped because …" The impact of the cmdlets that did not complete is inconclusive; static analysis and direct host access would be required to determine the true result.

Scope of Script Execution

Hosts

All script execution events occurred on the WS01.corp.local host.

```
index=inc06
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
         System.Security_attributes.UserID AS userid
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| rex field="Data.#text{}" "ScriptName=(?<scriptname>[^\n]+)"
| stats count, min(_time) as mintime, max(_time) as maxtime by computer
| convert ctime(mintime), ctime(maxtime)
```

computer	count	mintime	maxtime
DESKTOP-ET7845K	29	07/15/2026 15:25:00.416040	07/16/2026 01:24:49.489457
WIN-0Q9LS5QPQRJ	29	07/16/2026 01:20:30.203065	07/16/2026 01:24:24.107414
WS01	610	07/15/2026 16:23:04.987858	07/16/2026 21:32:33.203902
WS01.corp.local	16172	07/16/2026 11:48:07.694691	08/03/2026 14:51:57.859188

Figure 10: Host activity distribution.

```
index=inc06 latest="07/16/2026:11:48:07"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
         System.Security_attributes.UserID AS userid
| search computer!=*WS01*
| stats values(event) by computer
```

computer	values(event)
DESKTOP-ET7845K	2000
	2002
	21
	22
	23
	24
	32
	34
	39
	40
	41
	42
	5007
	54
WIN-0Q9LS5QPQRJ	32
	34
	5007
	54

Figure 11: Enumeration of DESKTOP-ET7845K and WIN-0Q9LS5QPQRJ showed no command execution activity.

```
index=inc06 latest="07/16/2026:11:48:07"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
| search computer=WS01
| stats count by event
```

event	count
1150	3
1151	3
2000	4
2010	2
21	4
22	4
23	3
32	5
41	4
42	4
5007	567
5016	1
54	5

Figure 12: All WS01 commands are unrelated to command execution, confirming no computer other than WS01.corp.local executed scripts.

Confidence: Low. The chronological progression of host activity is consistent with a new Windows installation being renamed to WS01 and joined to the CORP domain. This interpretation is not confirmed by the available logs and would need deeper analysis to verify.

Scripts

Searching for all executed PowerShell scripts across the environment shows the scale of the attack.

```
index=inc06
| rename System.EventID as event
| rex field="EventData.Data.#text{}" "ScriptName=(?<script>\S+)"
| rex field="EventData.ContextInfo" "Script\sName\s=\s(?<script1>\S+)"
| search event IN (800, 4103)
| eval script = coalesce(script, script1)
| dedup script
| table script
```

script ⇕
Z:\coll_scripts\log_collection.ps1
C:\Users\cyberlabs\Downloads\ransomware.ps1
C:\Windows\system32\WindowsPowerShell\v1.0\Modules\BitLocker\en-US\BitLocker.psd1
C:\Program
C:\Users\johnsmith\Desktop\Clear-Windows-LabLogs.ps1
C:\Windows\system32\WindowsPowerShell\v1.0\Modules\NetTCPIP\Test-NetConnection.psm1
C:\Windows\system32\WindowsPowerShell\v1.0\Modules\NetTCPIP\NetIPConfiguration.psm1
C:\Users\johnsmith\Downloads\Generate-NormalActivity.ps1

Figure 13: All scripts executed in the environment, captured via the coalesced 'ScriptName'/'Script Name' field of 800 & 4103 events. Beyond ransomware.ps1, several notable scripts are present: log_collection (discovery), Clear-Windows-LabLogs and Generate-NormalActivity (obfuscation).

Connecting script execution to user sessions identifies other compromised domain accounts — itadmin, Administrator, and johnsmith.

```
index=inc06
| rename System.EventID as event
        System.Computer as computer
        EventData.* as *
        UserData.EventXML.Address AS address
        UserData.EventXML.User AS user
        System.Security_attributes.UserID AS userid
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| rex field="Data.#text{}" "ScriptName=(?<scriptname>[^\n]+)"
| search computer=WS01.corp.local
    AND event IN (800,4103)
    AND NOT userid IN (S-1-5-18, S-1-5-20)
| dedup userid scriptname
| table userid scriptname
```

userid	scriptname
CORP\itadmin	Z:\coll_scripts\log_collection.ps1
CORP\itadmin	
CORP\cyberlabs	
CORP\cyberlabs	C:\Users\cyberlabs\Downloads\ransomware.ps1
CORP\cyberlabs	C:\Windows\system32\WindowsPowerShell\v1.0\Modules\BitLocker\en-US\BitLocker.psd1
CORP\SYSTEM	
CORP\itadmin	C:\Windows\system32\WindowsPowerShell\v1.0\Modules\BitLocker\en-US\BitLocker.psd1
CORP\itadmin	C:\Program Files\WindowsPowerShell\Modules\PSReadline\2.0.0\PSReadLine.psm1
CORP\itadmin	C:\Users\johnsmith\Desktop\Clear-Windows-LabLogs.ps1
CORP\johnsmith	
CORP\johnsmith	C:\Program Files\WindowsPowerShell\Modules\PSReadline\2.0.0\PSReadLine.psm1
CORP\itadmin	C:\Windows\system32\WindowsPowerShell\v1.0\Modules\NetTCPIP\Test-NetConnection.psm1
CORP\itadmin	C:\Windows\system32\WindowsPowerShell\v1.0\Modules\NetTCPIP\NetIPConfiguration.psm1
CORP\Administrator	
CORP\Administrator	C:\Program Files\WindowsPowerShell\Modules\PSReadline\2.0.0\PSReadLine.psm1
CORP\johnsmith	C:\Users\johnsmith\Downloads\Generate-NormalActivity.ps1

Figure 14: Focused queries to remove system user noise. Image shows script execution by multiple users. itadmin, cyberlabs, and johnsmith are seen running scripts of interest; definitive evidence of johnsmith's and Administrator's involvement appears in Figures 17 and 19. No other computer in the environment ran scripts (see "Scope of Script Execution"), so this list is comprehensive based on the fields captured.

The HostApplication field shows additional information regarding script execution activity.

```
index=inc06 latest="08/03/2026:14:47:33"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Param1 AS username
         UserData.EventXML.Param3 AS sourceip
| rex field="Data.#text{}" "CommandLine=(?<command>[^\n]+)"
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| rex field="Data.#text{}" "HostApplication=(?<hostapplication>[^\n]+)"
| rex field="Data.#text{}" "SequenceNumber=(?<sequencenumber>[^\n]+)"
| search computer=WS01.corp.local event=800
| stats values(command), min(_time) as mintime, max(_time) as maxtime by hostapplication userid
| convert ctime(mintime), ctime(maxtime)
```

powershell.exe -ExecutionPolicy Bypass new_domain_user.ps1		CORP\itadmin	08/03/2026 14:46:59.535077
CORP\itadmin	Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Add-MpPreference' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Get-MpBehavioralNetworkBlockingRules' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Get-MpComputerStatus' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Get-MpPreference' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Get-MpThreat' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Get-MpThreatCatalog' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Get-MpThreatDetection' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Remove-MpBehavioralNetworkBlockingRules' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Remove-MpPreference' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Remove-MpThreat' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Set-MpPreference' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Start-MpRollback' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Start-MpScan' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Start-MpWDOScan' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function 'Update-MpSignature' -Alias '*' Microsoft.PowerShell.Core\Export-ModuleMember -Function @() Set-MpPreference -DisableRealtimeMonitoring $true -ErrorAction Stop try { Microsoft.PowerShell.Core\Set-StrictMode -Off } catch { }		08/03/2026 14:05:10.016898

Figure 15: Filtered results of the query, focused on the execution of new-domain-user.ps1 and the disabling Windows Defender by itadmin, prior to the start of the cyberlabs session. The creation of a new user happened seconds before the cyberlabs session. This is strong evidence that this script created the cyberlabs account, which was later used to run ransomware.ps1.

Users

Johnsmith

Comparing script execution times across users shows the johnsmith account was the first to execute scripts of interest.

```
index=inc06
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
         System.Security_attributes.UserID AS userid
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| search computer=WS01.corp.local AND event IN (4103, 800) AND NOT userid IN (S-1-5-18, S-1-5-20)
| stats min(_time) as mintime, max(_time) as maxtime, count by date_mday, date_month, event, userid
| convert ctime(mintime), ctime(maxtime)
```

date_mday	date_month	event	userid	mintime	maxtime	count
16	july	800	CORP\Administrator	07/16/2026 13:50:19.349223	07/16/2026 14:10:17.229305	32
16	july	800	CORP\SYSTEM	07/16/2026 13:15:42.882211	07/16/2026 13:15:42.882211	2
19	july	800	CORP\SYSTEM	07/19/2026 12:15:53.801651	07/19/2026 14:14:40.568428	4
20	july	800	CORP\SYSTEM	07/20/2026 14:21:51.144638	07/20/2026 14:21:51.144638	2
21	july	800	CORP\SYSTEM	07/21/2026 13:12:32.438344	07/21/2026 13:12:32.438344	2
22	july	800	CORP\SYSTEM	07/22/2026 13:56:59.339372	07/22/2026 13:56:59.339372	2
23	july	800	CORP\SYSTEM	07/23/2026 13:47:54.621913	07/23/2026 13:47:54.621913	2
3	august	800	CORP\SYSTEM	08/03/2026 14:31:32.478323	08/03/2026 14:50:09.614912	12
31	july	800	CORP\SYSTEM	07/31/2026 13:40:23.189115	07/31/2026 13:40:25.001284	6
3	august	800	CORP\cyberlabs	08/03/2026 14:50:45.240924	08/03/2026 14:50:51.955946	3143
1	august	800	CORP\itadmin	08/01/2026 10:53:14.340557	08/01/2026 10:53:51.512770	59
3	august	800	CORP\itadmin	08/03/2026 09:42:52.930801	08/03/2026 14:51:57.859188	863
31	july	800	CORP\itadmin	07/31/2026 13:39:29.235463	07/31/2026 13:46:20.000414	264
16	july	800	CORP\johnsmith	07/16/2026 12:09:27.568391	07/16/2026 12:12:22.035863	236
3	august	800	CORP\johnsmith	08/03/2026 09:42:33.887841	08/03/2026 09:42:33.997081	2
31	july	800	CORP\johnsmith	07/31/2026 13:39:07.142191	07/31/2026 13:39:10.908200	3
3	august	4103	S-1-5-21-4260785253-2099098590-1992057404-1105	08/03/2026 14:51:49.377358	08/03/2026 14:51:54.661855	376
3	august	4103	S-1-5-21-4260785253-2099098590-1992057404-1126	08/03/2026 14:50:46.932373	08/03/2026 14:50:51.962428	2543

Figure 16: The first script execution session is on July 16th, 12:09, by user johnsmith (5th row from the end).

The user johnsmith ran discovery PowerShell commands:

```
index=inc06 latest="08/03/2026:14:47:33"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Param1 AS username
         UserData.EventXML.Param3 AS sourceip
| rex field="Data.#text{}" "CommandLine=(?<command>[^\n]+)"
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| rex field="Data.#text{}" "HostApplication=(?<hostapplication>[^\n]+)"
| rex field="Data.#text{}" "SequenceNumber=(?<sequencenumber>[^\n]+)"
| search computer=WS01.corp.local event=800
| stats values(command), min(_time) as mintime, max(_time) as maxtime by hostapplication userid
| convert ctime(mintime), ctime(maxtime)
```

hostapplication	userid	values(command)	mintime	maxtime

PowerShell.exe -noexit -command Set-Location -literalPath 'C:\Users\johnsmith\Downloads'	CORP\johnsmith	$dest = Join-Path $root $selected.Name
		$selected = Get-Random $files
		Copy-Item $selected.FullName $dest -Force
		Get-Item $dest | Out-Null
		$files = Get-ChildItem $path -File -Recurse | Select-Object -First 20
		'ipconfig' { ipconfig /all | Out-Null }
		'notepad' { $p=Start-Process notepad.exe -PassThru; Start-Sleep 3; Stop-Process $p.Id -Force
		}
		$choice = Get-Random @('notepad','calc','whoami','hostname','ipconfig','tasklist')
		$copy = Join-Path $desktop "WorkNotes-$stamp.txt"
		$f = Join-Path $root "WorkNotes-$stamp.txt"
		$line = "$(Get-Date -Format 'yyyy-MM-dd HH:mm:ss')
		[$env:USERDOMAIN\$env:USERNAME@$env:COMPUTERNAME] $Message"
		$share = Get-Random $ShareNames
		$stamp = Get-Date -Format 'yyyyMMdd-HHmmss'
		$t = Get-Random $targets
		@('Routine work notes',"User: $env:USERDOMAIN\$env:USERNAME","Computer:
		$env:COMPUTERNAME","Created: $(Get-Date -Format o)") | Set-Content $f
		Add-Content $f "Updated: $(Get-Date -Format o)"
		Add-Content -Path $log -Value $line
		Copy-Item $f $copy -Force
		Get-ChildItem $env:USERPROFILE -Directory | Out-Null
		Get-Content $f | Out-Null
		Get-Date | Out-Null
		Get-Process | Sort-Object CPU -Descending | Select-Object -First 5 | Out-Null
		Get-Service | Where-Object Status -eq Running | Select-Object -First 10 | Out-Null
		Microsoft.PowerShell.Core\Set-StrictMode -Off
		Rename-Item $copy "Reviewed-WorkNotes-$stamp.txt" -Force
		Resolve-DnsName $t | Out-Null
		Write-Host $line
		if (Test-Path $path) {
		switch (Get-Random $actions) {
		Write-Host "Output folder: $root"
		function Pause-Random { Start-Sleep -Seconds (Get-Random -Minimum 4 -Maximum 15) }
		while ((Get-Date) -lt $end) {
			07/16/2026 12:09:27.568391	07/16/2026 12:12:22.035863

Figure 17: On July 16th, 12:09, johnsmith executed a script containing discovery commands from the Downloads folder (selecting at random either whoami, hostname, ipconfig or tasklist), indicating that the script is malicious and the user/session is compromised.

The investigation continues by tracing the initial access to johnsmith.

```
index=inc06
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
         System.Security_attributes.UserID AS userid
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| rex field="Data.#text{}" "ScriptName=(?<scriptname>[^\n]+)"
| search computer=WS01.corp.local  johnsmith event=21
| table _time address user
```

_time	address	user
2026-08-01 10:50:57.272	LOCAL	CORP\johnsmith
2026-07-31 13:38:37.724	LOCAL	CORP\johnsmith
2026-07-16 11:49:59.096	LOCAL	CORP\johnsmith

Figure 18: The johnsmith logon was a local connection — a physical "keyboard and mouse" session.

Confidence: Low on johnsmith as true patient zero. Since the connection was local, the first johnsmith session was likely organic. Assuming johnsmith is a legitimate account belonging to a regular employee, two scenarios remain equally plausible: johnsmith is patient zero, or the host was compromised prior to the scope of the collected logs. Neither can be confirmed from the available evidence. What can be said with confidence is that the observed chain of compromise, within this log set, begins with johnsmith.

Administrator

The next session after the initial johnsmith session is by Administrator.

```
index=inc06 latest="08/03/2026:14:47:33"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Param1 AS username
         UserData.EventXML.Param3 AS sourceip
| rex field="Data.#text{}" "CommandLine=(?<command>[^\n]+)"
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| rex field="Data.#text{}" "HostApplication=(?<hostapplication>[^\n]+)"
| rex field="Data.#text{}" "SequenceNumber=(?<sequencenumber>[^\n]+)"
| search computer=WS01.corp.local event=800
| stats values(command), min(_time) as mintime, max(_time) as maxtime by hostapplication userid
| convert ctime(mintime), ctime(maxtime)
```

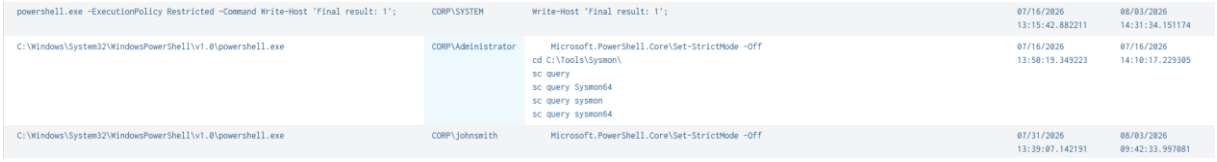

Figure 19: On July 16th, 13:50, Administrator executed commands to weaken PowerShell execution restrictions and query the Sysmon process — less than two hours after the malicious commands executed by johnsmith on the same host.

Confidence: Low. A full review of all command execution events by Administrator was not performed; the sampled commands in Figure 19 are consistent with an escalation of privileges from johnsmith to Administrator, but this rests on a small sample and is not independently confirmed.

itadmin

Figure 18 shows a logon session to johnsmith on July 31st around 13:38. One minute later, on the same host, itadmin runs a script which includes directory traversal, file deletion, and stopping Sysmon.

```
index=inc06 latest="08/03/2026:14:47:33"
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Param1 AS username
         UserData.EventXML.Param3 AS sourceip
| rex field="Data.#text{}" "CommandLine=(?<command>[^\n]+)"
| rex field="Data.#text{}" "UserId=(?<userid>[^\n]+)"
| rex field="Data.#text{}" "HostApplication=(?<hostapplication>[^\n]+)"
| rex field="Data.#text{}" "SequenceNumber=(?<sequencenumber>[^\n]+)"
| search computer=WS01.corp.local event=800
| stats values(command), min(_time) as mintime, max(_time) as maxtime by hostapplication userid
| convert ctime(mintime), ctime(maxtime)
```

```
CORP\itadmin                                    $errorCategoryMsg = & { Set-StrictMode -Version 1;              07/31/2026          08/03/2026
                $_.ErrorCategory_Message }                                                                      13:39:29.235463     09:43:40.212722
                                                $originInfo = & { Set-StrictMode -Version 1; $_.OriginInfo }
                                                if ( & { Set-StrictMode -Version 1; $_.PSMessageDetails } )
                {
                        $foundSuggestion = @(Get-Command ($ExecutionContext.SessionState.Path.Combine(".",
                $escapedCommand)) -ErrorAction Ignore).Count -gt 0
                    Microsoft.PowerShell.Core\Set-StrictMode -Off
                    Write-Host "Clearing $Log"
                & { Set-StrictMode -Version 1; $this.Exception.InnerException.PSMessageDetails }
                ConvertFrom-StringData -stringdata @'
                Enable-NetFirewallRule -DisplayGroup "Remote Desktop"
                Export-ModuleMember -Alias TNC -Function Test-NetConnection
                Export-ModuleMember -Alias gip -Function Get-NetIPConfiguration
                Get-NetTCPConnection -LocalPort 3389 -State
                Get-NetTCPConnection -LocalPort 3389 -State Listen
                New-Alias TNC Test-NetConnection
                New-Alias gip Get-NetIPConfiguration
                Remove-Item "$env:TEMP\*" -Recurse -Force -ErrorAction SilentlyContinue
                Remove-Item "C:\Windows\Temp\*" -Recurse -Force -ErrorAction SilentlyContinue
                Set-ItemProperty `
                Set-Service TermService -StartupType Automatic
                Start-Service Sysmon64 -ErrorAction SilentlyContinue
                Start-Service TermService
                Stop-Service Sysmon64 -ErrorAction SilentlyContinue
                Write-Host ""
                Write-Host "Deleting temp files..."
                Write-Host "Restarting Sysmon..."
                Write-Host "Stopping Sysmon..."
                Write-Host "Windows lab reset complete."
                cd ..
                cd .\Desktop\
                cd .\Users\
                cd .\johnsmith\
                cd C:\Users\johnsmith\Desktop\
                dir
                try { Microsoft.PowerShell.Core\Set-StrictMode -Off } catch { }
```

Figure 20: At 13:39, itadmin executes a script which includes directory traversal, file deletion, stopping the Sysmon process (consistent with the Administrator session in Figure 19), and enabling a firewall rule to allow RDP connections.

Six minutes later, a successful RDP connection was registered to itadmin (Figure 6). The logs do not show how the attacker obtained access to itadmin or to the host prior to this point.

Examination of event codes generated by itadmin shows Windows Defender malicious code detection.

```
index=inc06 itadmin
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
         System.Security_attributes.UserID AS userid
| stats values(event), count by date_mday
```

values(event)	count
800	59
1116 1117 21 22 24 25 41 4103 42 800	1259
1149 41 800	266

Figure 21: 1116 and 1117 events for itadmin, indicating malicious code detection and response by Windows Defender.

Windows Defender logs show itadmin attempting to execute a file named `payload.exe` from within the AppData folder, which is quarantined and flagged as a Meterpreter trojan. This occurred less than an hour before the cyberlabs session.

```
index=inc06 itadmin
| rename System.EventID as event
         System.Computer as computer
         EventData.* as *
         UserData.EventXML.Address AS address
         UserData.EventXML.User AS user
         System.Security_attributes.UserID AS userid
| search event IN (1116, 1117)
| table _time "Action Name" "Category Name" "Detection User" Path "Threat Name"
```

_time	Action Name	Category Name	Detection User	Path	Threat Name
2026-08-03 14:31:24.199	Quarantine	Trojan	CORP\itadmin	file:_C:\Users\itadmin\AppData\Local\Microsoft\Windows\INetCache\IE\1N0ICO2Y\payload[1].exe; file:_C:\Windows\Temp\payload.exe	Trojan:Win32/Meterpreter.RPZ!MTB
2026-08-03 14:05:38.410	Not Applicable	Trojan	CORP\itadmin	file:_C:\Users\itadmin\AppData\Local\Microsoft\Windows\INetCache\IE\1N0ICO2Y\payload[1].exe; file:_C:\Windows\Temp\payload.exe	Trojan:Win32/Meterpreter.RPZ!MTB
2026-08-03 14:05:38.382	Quarantine	Trojan	CORP\itadmin	file:_C:\Users\itadmin\AppData\Local\Microsoft\Windows\INetCache\IE\1N0ICO2Y\payload[1].exe	Trojan:Win32/Meterpreter.RPZ!MTB
2026-08-03 14:05:25.834	Not Applicable	Trojan	CORP\itadmin	file:_C:\Users\itadmin\AppData\Local\Microsoft\Windows\INetCache\IE\1N0ICO2Y\payload[1].exe; file:_C:\Windows\Temp\payload.exe	Trojan:Win32/Meterpreter.RPZ!MTB
2026-08-03 14:05:16.533	Not Applicable	Trojan	CORP\itadmin	file:_C:\Windows\Temp\payload.exe	Trojan:Win32/Meterpreter.RPZ!MTB
2026-08-03 14:05:09.644	Not Applicable	Trojan	CORP\itadmin	file:_C:\Users\itadmin\AppData\Local\Microsoft\Windows\INetCache\IE\1N0ICO2Y\payload[1].exe	Trojan:Win32/Meterpreter.RPZ!MTB

Figure 22: itadmin attempting to execute a Meterpreter payload. It is unclear from the logs whether he succeeded.

The full list of scripts which were found during the investigation to be executed by itadmin:

- `Z:\coll_scripts\log_collection.ps1`
- `C:\Users\johnsmith\Desktop\Clear-Windows-LabLogs.ps1`
- `Test-NetConnection.psm1`
- `NetIPConfiguration.psm1`
- `new-domain-user.ps1`
- `C:\Windows\Temp\payload.exe`

Confidence: Low on the completeness of this list beyond what was captured in the investigation. For example, other than the standard C: partition, a Z: partition is present containing a `coll_scripts` folder. It likely holds additional scripts not captured by the queries run here, and its contents should be reviewed in a follow-up analysis. In addition, a comprehensive static file analysis \ host artifact analysis is required to analyze all command executions to fully determine the scope of what itadmin executed during the attack cycle. In any case, this list is enough to get the big picture: itadmin is the main domain tool used by the attacker to facilitate their intentions.

Open Questions & Recommended Follow-up

- **Identity of source IP 10.0.2.7** — appears only in the two RDP handshake events (Figure 6); not resolved to a host, user, or external/internal origin.
- **Initial access vector for itadmin, johnsmith** — no log evidence was found to show how the attacker obtained credentials or host access.
- **WS01 host provisioning history** — the apparent OS install / domain-join sequence (Scope of Script Execution) is inferred from timing alone and is not confirmed by provisioning or AD-join logs. Even if a follow-up investigation confirms the suspicion that the attacker added this host to the domain, it remains unclear how he was able to do that, or what tools he later used to get credentail access to privileged users.
- **Contents of `Z:\coll_scripts\`** — the directory likely contains additional scripts relevant to the incident that were not captured by the SPL queries used in this investigation. Moreover, the existence of a non-standard disk partition (Z:\) is suspicious – validity should be verified with IT.
- **Full review of command execution events** — only a sample was reviewed; a complete review may shed new light on the investigation.
- **Patient zero determination for johnsmith** — cannot be distinguished from a prior, out-of-scope compromise of the host without additional log sources (e.g., endpoint history predating July 16th or other log source types).

Impact Assessment

Based on the available logs, the confirmed technical impact is confined to a single host, WS01.corp.local, and four domain accounts (johnsmith, Administrator, itadmin, cyberlabs). Within that scope, the attacker executed a script consistent with disk-encryption intent (BitLocker module invocation) and attempted to establish an additional foothold via a Meterpreter payload executed under itadmin. Whether the encryption activity completed successfully cannot be confirmed — cmdlet execution events are intertwined with error events over the same window, and this can only be resolved through static analysis and direct host access. The success of the attempted meterpreter session is also inconclusive and requires further examination.

The attacker also took steps to reduce visibility and hinder response: Sysmon and Windows Defender (AV) were stopped, PowerShell execution restrictions were weakened, and a log-clearing script was executed in the environment. This anti-forensic activity means the picture reconstructed here may be incomplete, particularly for the period before the first captured johnsmith session.

The scope of impact beyond WS01.corp.local is unconfirmed rather than ruled out. No script execution was observed on other hosts in the available logs, and the source IP behind the RDP connections (10.0.2.7) was not resolved to an internal origin. Until the initial access vector and that IP are identified, the possibility of a broader foothold elsewhere in the environment cannot be excluded.

Recommendations

Restore

- Isolate WS01.corp.local from the network pending forensic imaging and static analysis of PowerShell scripts and executables.
- Disable and reset credentials for the johnsmith, Administrator and itadmin accounts.
- Remove the cyberlabs account, created outside of normal provisioning via new-domain-user.ps1.
- Re-enable Sysmon and Windows Defender on WS01.corp.local and verify that their logging configuration has not been otherwise tampered with.
- Remove identified malicious artifacts from the host.
- If encryption is confirmed on affected volumes, restore the most recent verified pre-compromise backup.

Prevent

- Enforce MFA on RDP and all privileged account logons.
- Restrict RDP access at the network/firewall level; alert on unauthorized firewall rule changes enabling RDP.
- Review and consider reducing the standing privileges for johnsmith; apply least-privilege access controls.
- Enable PowerShell script block logging and constrained language mode across the environment, not just on WS01.
- Add alerts for events that indicate tampering with the Sysmon and Windows Defender services.
- Review and inventory the contents of `Z:\coll_scripts`.
- Investigate the initial johnsmith compromise and the identity of source IP 10.0.2.7 to close the open initial-access question.

Conclusion

The investigation reconstructs a plausible, evidence-based chain of compromise on WS01.corp.local, progressing from johnsmith through Administrator and itadmin to the attacker-created cyberlabs account, which was used to execute a script consistent with ransomware/disk-encryption behavior. Several elements of this chain rest on strong, cross-referenced evidence — particularly the link between cyberlabs, ransomware.ps1, and the RDP session — while others, including the initial access vector, the success of the encryption attempt, and the identity behind source IP 10.0.2.7, remain open and are documented above as follow-up items. The findings support treating WS01.corp.local and all four identified accounts as compromised and prioritizing containment, credential resets, and further forensic review before returning the host to service.