

Detecting Cobalt Strike's PSEXEC

Cobalt Strike's `psexec` command is an implementation of the popular PsExec tool, which is a part of Microsoft's Sysinternals Suite. It's a lightweight telnet-replacement that lets you execute processes on other systems. Cobalt Strike's version is utilized to execute payloads on remote systems, as part of the post-exploitation process.

When the `psexec` command is invoked within Cobalt Strike, the following steps occur:

- Service Creation:** The tool first creates a new service on the target system. This service is responsible for executing the desired payload. The service is typically created with a random name to avoid easy detection.
- File Transfer:** Cobalt Strike then transfers the payload to the target system, often to the `ADMIN$` share. This is typically done using the SMB protocol.
- Service Execution:** The newly created service is then started, which in turn executes the payload. This payload can be a shellcode, an executable, or any other file type that can be executed.
- Service Removal:** After the payload has been executed, the service is stopped and deleted from the target system to minimize traces of the intrusion.
- Communication:** If the payload is a beacon or another type of backdoor, it will typically establish communication back to the Cobalt Strike team server, allowing for further commands to be sent and executed on the compromised system.

Cobalt Strike's `psexec` works over port 445 (SMB), and it requires local administrator privileges on the target system.

Therefore, it's often used after initial access has been achieved and privileges have been escalated.

How Cobalt Strike PSEXEC Traffic Looks Like

Protocol	Length	Info
SHB2	655	Session Setup Request, NTLMSSP_AUTH, User: WIN10\vagrant
SHB2	159	Session Setup Response
SHB2	154	Tree Connect Request Tree: \\WeF\ADMIN\$
SHB2	138	Tree Connect Response
SHB2	178	Ioctl Request FSCTL_QUERY_NETWORK_INTERFACE_INFO
SHB2	474	Ioctl Response FSCTL_QUERY_NETWORK_INTERFACE_INFO
SHB2	234	Create Request File:
SHB2	298	Create Response File:
SHB2	146	Close Request File:
SHB2	182	Close Response
SHB2	382	Create Request File: c3400ec.exe
SHB2	410	Create Response File: c3400ec.exe
SHB2	39474	Write Request Len:65536 Off:0 File: c3400ec.exe [TCP segment of a reassembled PDU]
SHB2	138	Write Response
SHB2	481	Write Request Len:65536 Off:65536 File: c3400ec.exe
SHB2	138	Write Response
SHB2	1466	Write Request Len:65536 Off:131072 File: c3400ec.exe
SHB2	138	Write Response
SHB2	1466	Write Request Len:65536 Off:196608 File: c3400ec.exe
SHB2	20659	Write Request Len:20480 Off:262144 File: c3400ec.exe
SHB2	138	Write Response
SHB2	138	Write Response
SHB2	3242	Write Request Len:3072 Off:282624 File: c3400ec.exe
SHB2	138	Write Response
SHB2	162	GetInfo Request FILE_INFO/SMB2_FILE_NETWORK_OPEN_INFO File: c3400ec.exe
SHB2	186	GetInfo Response
SHB2	146	Close Request File: c3400ec.exe
SHB2	182	Close Response
SHB2	126	Tree Disconnect Request
SHB2	126	Tree Disconnect Response
SHB2	126	Session Logoff Request
SHB2	126	Session Logoff Response
SHB2	272	Negotiate Protocol Request
SHB2	366	Negotiate Protocol Response
SHB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
SHB2	395	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
SHB2	655	Session Setup Request, NTLMSSP_AUTH, User: WIN10\vagrant
SHB2	159	Session Setup Response
SHB2	159	Tree Connect Request Tree: \\WeF\IPC\$
SHB2	138	Tree Connect Response
SHB2	178	Ioctl Request FSCTL_QUERY_NETWORK_INTERFACE_INFO
SHB2	190	Create Request File: svccntl
SHB2	474	Ioctl Response FSCTL_QUERY_NETWORK_INTERFACE_INFO
SHB2	210	Create Response File: svccntl
SHB2	162	GetInfo Request FILE_INFO/SMB2_FILE_STANDARD_INFO File: svccntl
SHB2	154	GetInfo Response
DCERPC	286	Bind: call_id: 2, Fragment: Single, 2 context items: SVCCTL V2.0 (32bit NDR), SVCCTL V2.0 (6cb71c2c-9812-4540-0300-000000000000)
SHB2	138	Write Response
SHB2	171	Read Request Len:1024 Off:0 File: svccntl
DCERPC	230	Bind_ack: call_id: 2, Fragment: Single, max_xmit: 4280 max_recv: 4280, 2 results: Acceptance, Negotiate ACK
SVCCTL	210	Unknown operation 64 request
DCERPC	202	Fault: call_id: 2, Fragment: Single, Ctx: 0, status: nca_op_rng_error
CURRENT	2	MANIFESTREQUEST: Win32ManifestRequest

Resources

Go to Questions

Table of Contents

Leveraging Windows Event Logs

Detecting Common User/Domain Recon	✓
Detecting Password Spraying	✓
Detecting Responder-like Attacks	✓
Detecting Kerberoasting/AS-REPROasting	✓
Detecting Pass-the-Hash	✓
Detecting Pass-the-Ticket	✓
Detecting Overpass-the-Hash	✓
Detecting Golden Tickets/Silver Tickets	✓
Detecting Unconstrained Delegation/Constrained Delegation Attacks	✓
Detecting DCSync/DCShadow	✓

Leveraging Splunk's Application Capabilities

Creating Custom Splunk Applications	✓
-------------------------------------	---

Leveraging Zeek Logs

Detecting RDP Brute Force Attacks	✓
Detecting Beacons Malware	✓
Detecting Nmap Port Scanning	✓
Detecting Kerberos Brute Force Attacks	✓
Detecting Kerberoasting	✓
Detecting Golden Tickets	✓
Detecting Cobalt Strike's PSEXEC	✓
Detecting Zerologon	✓
Detecting Exfiltration (HTTP)	✓
Detecting Exfiltration (DNS)	✓
Detecting Ransomware	✓

Skills Assessment

Image Source: <https://thedfirereport.com/2021/08/29/cobalt-strike-a-defenders-guide/>

Let's now navigate to the bottom of this section and click on "Click here to spawn the target system!". Then, access the Splunk interface at [https://\[Target IP\]:8000](https://[Target IP]:8000) and launch the Search & Reporting Splunk application. The vast majority of searches covered from this point up to end of this section can be replicated inside the target, offering a more comprehensive grasp of the topics presented.

Additionally, we can access the spawned target via RDP as outlined below. All files, logs, and PCAP files related to the covered attacks can be found in the /home/htb-student and /home/htb-student/module_files directories.

Related Evidence

- Related Directory: `/home/htb-student/module_files/cobalt_strike_psexec`
- Related Splunk Index: `cobalt_strike_psexec`
- Related Splunk Sourcetype: `bro:smb_files:json`

Detecting Cobalt Strike's PSEXEC With Splunk & Zeek Logs

Now let's explore how we can identify Cobalt Strike's PSEXEC, using Splunk and Zeek logs.

i	Time	Event
>	8/22/21 7:00:41.000 AM	{ [-] action: SMB::FILE_OPEN id.orig_h: 192.168.38.104 id.orig_p: 49394 id.resp_h: 192.168.38.102 id.resp_p: 445 name: be5312f.exe path: \\DC\\ADMIN\$ size: 285696 times.accessed: 2021-08-22T07:00:20.467586Z times.changed: 2021-08-22T07:00:20.475082Z times.created: 2021-08-22T07:00:20.467586Z times.modified: 2021-08-22T07:00:20.475082Z ts: 2021-08-22T07:00:41.577519Z uid: CfU6Kn6isF0IrfxL }

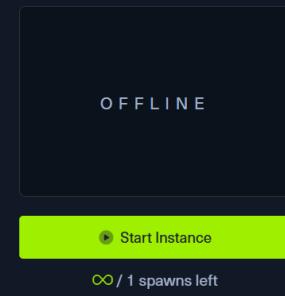
VPN Servers

⚠ Warning: Each time you "Switch", your connection keys are regenerated and you must re-download your VPN connection file.

All VM instances associated with the old VPN Server will be terminated when switching to a new VPN server.

Skills Assessment

My Workstation



Existing PwnBox Instances will automatically switch to the new VPN server.

US Academy 3

Medium Load ▾

PROTOCOL

UDP 1337 TCP 443

DOWNLOAD VPN CONNECTION FILE



Connect to Pwnbox

Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK

136ms ▾

Terminate Pwnbox to switch location

Start Instance

∞ / 1 spawns left

Waiting to start...



Enable step-by-step solutions for all questions

Questions

Answer the question(s) below to complete this Section and earn cubes!



Download VPN
Connection File

Target(s): [Click here to spawn the target system!](#)

+ 1 Use the "change_service_config" index and the "bro:dce_rpc:json" sourcetype to create a Splunk search that will detect SharpNoPSEExec (<https://gist.github.com/defensivedepth/ae3f882efa47e20990bc562a8b052984>). Enter the IP included in the "id.orig_h" field as your answer.

192.168.38.104

Submit

◀ Previous

Next ▶

Mark Complete & Next

Powered by  HACKTHEBOX