New PsExec spinoff lets hackers bypass network security defenses

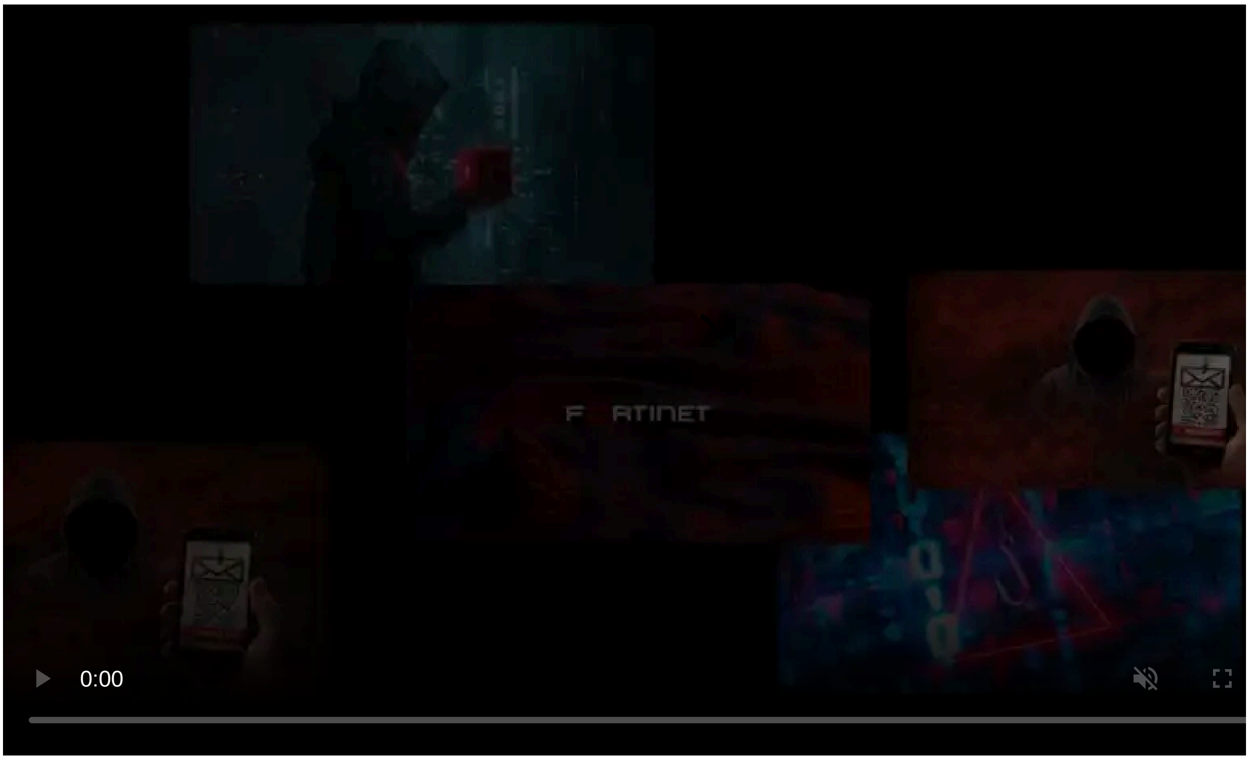
By Ionut Ilascu

Published: 2022-09-13 · Archived: 2026-04-06 02:51:49 UTC

Security researchers have developed an implementation of the Sysinternals PsExec utility that allows moving laterally in a network using a single, less monitored port, Windows TCP port 135.

PsExec is designed to help administrators execute processes remotely on machines in the network without the need to install a client.

Threat actors have also adopted the tool and are frequently using it in post-exploitation stages of an attack to spread on the network, run commands on multiple systems, or deploy malware.


0:00

PsExec and the TCP ports it needs

While the original PsExec is available in the Sysinternals utility suite, there is also an implementation in the Impacket collection of Python classes for working with network protocols, which has support for SMB and other protocols like IP, UDP, TCP that enable connections for HTTP, LDAP (Lightweight Directory Access Protocol), and Microsoft SQL Server (MSSQL).

Both the original version and the Impacket variant work in a similar way. They use an SMB connection and are based on port 445, which needs to be open to communicate over the SMB network file-sharing protocol.

They also manage Windows services (create, execute, start, stop) through Remote Procedure Calls (RPC), a protocol that enables high-level communication with the operating system.

For extended functionality, though, port 135 is required. However, blocking this port does not prevent a threat actor from completing an attack, therefore port 445 is essential for PsExec to work.

Because of this, defenders mostly focus on blocking port 445, which is essential for PsExec to execute commands or run files. This works in most cases but is not enough.

New PsExec implementation

Based on the Impacket library, researchers at Pentera, a company that provides an automated security validation solution, have built an implementation of the PsExec tool that runs only on port 135.

This achievement brings changes to the defense game since blocking just port 445 to restrict malicious PsExec activity is no longer a reliable option for most attacks.

“We found that the SMB protocol is used to upload the binary and to forward the input and output,” Yuval Lazar, a senior security researcher at Pentera explains.

Lazar adds in a report shared with BleepingComputer that commands are executed through Distributed Computing Environment / Remote Procedure Calls (DCE/RPC) and processes “run regardless of the output.”

Port	Protocol	Info
135	DCERPC	Bind: call_id: 1, Fragment: Single, 1 context items: EPMv4 V3.0 (32bit NDR)
55922	DCERPC	Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_recv: 4280, 1 results: Acceptan…
135	EPM	Map request, SVCCTL, 32bit NDR
55922	EPM	Map response, SVCCTL, 32bit NDR
49155	DCERPC	Bind: call_id: 1, Fragment: Single, 1 context items: SVCCTL V2.0 (32bit NDR), NTLMSSP_NEGO…
59598	DCERPC	Bind_ack: call_id: 1, Fragment: Single, max_xmit: 4280 max_recv: 4280, 1 results: Acceptan…
49155	DCERPC	AUTH3: call_id: 1, Fragment: Single, NTLMSSP_AUTH, User: WIN-PHS40F0ELBU\yuval
49155	SVCCTL	OpenSCManagerW request
59598	SVCCTL	OpenSCManagerW response
49155	SVCCTL	CreateServiceW request
59598	SVCCTL	CreateServiceW response
49155	SVCCTL	StartServiceW request
59598	SVCCTL	StartServiceW response

Running PsExec commands over port 135

source: Pentera Labs

The PsExec variation from Pentera uses an RPC connection that enabled the researchers to create a service that runs an arbitrary command without communicating over SMB port 445 for transport or output.

```python
def setDCERPCConnection(self, binding):

    rpc = transport.DCERPCTransportFactory(binding)
    rpc.set_credentials(username=self.__username, password=self.__password,
                        domain=self.__domain, nthash=self.__nthash)
    rpc.setRemoteHost(self.remoteHost)
    rpc.setRemoteName(self.remoteName)
    if self.__doKerberos:
        #using kerberos must include using hostname and full FQDN in domain
        rpc.set_kerberos(True, self.__dcip)

    try:
        LOG.info(f"Starting DCERPC connection to {self.remoteHost}")
        rpcsvc = rpc.get_dce_rpc()
        rpcsvc.set_credentials(*rpc.get_credentials())
        rpcsvc.set_auth_type(RPC_C_AUTHN_WINNT)
        rpcsvc.set_auth_level(RPC_C_AUTHN_LEVEL_PKT_PRIVACY)
        if self.__doKerberos:
            rpcsvc.set_auth_type(RPC_C_AUTHN_GSS_NEGOTIATE)
        rpcsvc.connect()
        rpcsvc.bind(scmr.MSRPC_UUID_SCMR)
        LOG.info(f"Successfully connected to {self.remoteHost}/MSRPC_UUID_SCMR")
        return rpcsvc
```

Pentera's PsExec implementation creates DCE/RPC connection without SMB

source: Pentera Labs

All-out monitoring needed

Unlike the original PsExec in the Sysinternals suite, Pentera’s variant has a higher chance of slipping undetected in a network, Lazar told BleepingComputer, because many organizations keep an eye on port 445 and SMB.

“What we’ve noticed is that while many organizations implement a lot of the mitigations based on SMB and port 445, they overlook other important ports such as 135” - Yuval Lazar, Senior Security Researcher at Pentera

Another point Lazar makes is that other PsExec implementations have to use SMB because they are file-based. Pentera’s variant is fileless, the researcher said, which would make it more difficult to detect.

Lazar’s research on PsExec highlights that while security vulnerabilities like PetitPotam [1, 2] and DFSCoerce have drawn attention to the risk RPC poses, mitigations don’t emphasize monitoring DCE/RPC but on NTLM relay prevention.

Based on Pentera’s observations, blocking or monitoring RPC traffic is not common practice in corporate environments. The reason in many cases is that defenders are unaware that RPC can introduce a security risk to the network if left unchecked.

“Security teams need to understand how different ports can be used by hackers so that they know what to monitor them for” - Yuval Lazar

Will Dormann, senior vulnerability analyst at ANALYGENCE, agrees that blocking TCP port 445 alone is insufficient to block malicious activity relying on the tool.

"If people think that blocking 445 only is enough to prevent PsExec (and other RPC-related things), then they are mistaken," the researcher told BleepingComputer.

PsExec is based on SMB and RPC connections, which require ports 445, 139, and 135. However, Lazar added that there is an RPC implementation on top of HTTP, meaning that PsExec could potentially work over port 80, too.

PsExec popular with ransomware actors

Hackers have been using PsExec in their attacks for a long time. Ransomware gangs, in particular, adopted it to deploy file-encrypting malware.

In an attack that lasted just one hour, NetWalker ransomware used PsExec to run their payload on all systems in a domain.

In a more recent example, the Quantum ransomware gang relied on PsExec and WMI to encrypt systems in an attack that took only two hours to complete after gaining access via IcedID malware.

A report from Microsoft in June details an attack from BlackCat ransomware, who also used PsExec to distribute their ransomware payload.

Another example is from the recently disclosed [Cisco breach](), where the Yanluowang ransomware gang used PsExec to add registry values remotely, allowing the threat actor to leverage the accessibility features available on the Windows logon screen.

Update [September 13, 10:10 EST]: *Article updated with comment from Will Dormann, vulnerability analyst at the U.S. CERT Coordination Center.*

Source: https://www.bleepingcomputer.com/news/security/new-psexec-spinoff-lets-hackers-bypass-network-security-defenses/