A glimpse into the shadowy realm of a Chinese APT: detailed analysis of a ShadowPad intrusion

: 9/30/2022

Authors: **William Backhouse** (@Will0x04), **Michael Mullen** (@DropTheBase64) and **Nikolaos Pantazopoulos**

Summary

tl;dr

This post explores some of the TTPs employed by a threat actor who was observed deploying ShadowPad during an incident response engagement.

Below provides a summary of findings which are presented in this blog post:

- Initial access via CVE-2022-29464.
- Successive backdoors installed – PoisonIvy, a previously undocumented backdoor and finally ShadowPad.
- Establishing persistence via Windows Services to execute legitimate binaries which sideloads backdoors, including ShadowPad.
- Use of information gathering tools such as ADFind and PowerView.
- Lateral movement leveraging RDP and ShadowPad.
- Use of 7zip for data collection.
- ShadowPad used for Command and Control.
- Exfiltration of data.

ShadowPad

This blog looks to build on the work of other security research done by SecureWorks and PwC with firsthand experience of TTPs used in a recent incident where ShadowPad was deployed. ShadowPad is a modular remote access trojan (RAT) which is thought to be used almost exclusively by China-Based threat actors.

Attribution

Based on the findings of our Incident Response investigation, NCC Group assesses with high confidence that the threat actor detailed in this article was a China-based Advanced Persistent Threat (APT).

This is based on the following factors

- ShadowPad – Public reporting has previously indicated the distribution of ShadowPad is tightly controlled and is typically exclusive to China-based threat actors for use during espionage campaigns.
- TTPs – Specific TTPs observed during the attack were found to match those previously observed by China-based threat actors, both within NCC Group incident response engagements and the wider security community.
- Activity pattern analysis – The threat actor was typically active during the hours of 01:00 – 09:00 (UTC) which matches the working hours of China

TTPs

Initial Access

A recent vulnerability in WSO2, CVE-2022-29464 [3], was the root cause of the incident. The actor, amongst other attackers, was able to exploit the vulnerability soon after it was published to create web shells on a server.

The actor leveraged a web shell to load a backdoor, in this case PoisonIvy. This was deployed via a malicious DLL and leveraged DLL Search Order Hijacking, a tactic which was continuously leveraged throughout the attack.

Execution

Certutil.exe was used via commands issued on web shells to install the PoisonIvy backdoor on patient zero.

The threat actor leveraged command prompt and PowerShell throughout the incident.

Additionally, several folders named _MEI<*random digits*> were observed within the Windows\Temp folder. The digits in the folder name change each time a binary is compiled. These folders are created on a host when a python executable is compiled. Within these folders were the .pyd library files and DLL files. The created time for these folders matched the last modified time stamp of the complied binary within the shimcache.

Persistence

Run Keys and Windows services were used throughout in order to ensure the backdoors deployed obtained persistence.

Defense Evasion

The threat actor undertook significant anti-forensic actions on ShadowPad related files to evade detection. This included timestomping the malicious DLL and applying the NTFS attributes of hidden and system to the files. Legitimate but renamed Windows binaries were used to load the configuration file. The threat actor also leveraged a legitimate Windows DLL, `secur32.dll`, as the name of the configuration file for the ShadowPad backdoor.

All indicators of compromise, aside from backdoor modules and loaders, were removed from the hosts by the threat actor.

Credential Access

The threat actor was observed collecting all web browser credentials from all hosts across the environment. It is unclear at this stage how this was achieved with the evidence available.

Discovery

A vast array of tooling was used to scan and enumerate the network as the actor negotiated their way through it, these included but were not limited to the following:

- AdFind
- NbtScan
- PowerView
- PowerShell scripts to enumerate hosts on port 445
- Tree.exe

Lateral Movement

Lateral movement was largely carried out using Windows services, particularly leveraging SMB pipes. The only interactive sessions observed were onward RDP sessions to customer connected sites.

Collection

In addition to the automated collection of harvested credentials, the ShadowPad keylogger module was used in the attack, storing the keystrokes in encrypted database files for exfiltration. The output of which was likely included in archive files created by the attacker, along with the output of network scanning and reconnaissance.

Command and Control

In total, three separate command and control infrastructures were identified, all of which utilised DLL search order hijacking / DLL side loading. The initial payload was PoisonIvy, this was only observed on patient zero. The threat actor went on to deploy a previously undocumented backdoor once they gained an initial foothold in the network, this framework established persistence via a service called K7AVWScn, masquerading as an older anti-virus product. Finally, once a firm foothold was established within the network the threat actor deployed ShadowPad. Notably, the ShadowPad module for the proxy feature was also observed during the attack to proxy C2 communications via a less conspicuous server.

Exfiltration

Due to the exfiltration capabilities of ShadowPad, it is highly likely to have been the method of exfiltration to steal data from the customer network. This is further cemented by a small, yet noticeable spike in network traffic to threat actor controlled infrastructure.

Recommendations

- Searches for the documented IOCs should be conducted
- If IOCs are identified a full incident response investigation should be conducted

ShadowPad Technical Analysis

Initialisation phase

Upon execution, the ShadowPad core module enters an initialisation phase at which it decrypts its configuration and determines which mode it runs. In summary, we identified the following modes:

Mode ID	Description
3	Injects itself to a specified process (specified in the ShadowPad configuration) and adds persistence to the compromised host. In addition, if the compromised user belongs to a group with a SID starting with S-1-5-80- then the specified target process uses the token of 'lsass'.
4	Injects itself to a specified process (specified in the ShadowPad configuration) and executes the core code in a new thread. In addition, if the compromised user belongs to a group with a SID starting with S-1-5-80 then the specified target process uses the token of 'lsass'.
5	Injects itself to a specified process (specified in the ShadowPad configuration). In addition, if the compromised user belongs to a group with a SID starting with S-1-5-80 then the specified target process uses the token of 'lsass'.
16	Injects itself to a specified process (specified in the ShadowPad configuration) and creates/starts a new service (details are specified in the ShadowPad configuration), which executes the core code. In addition, if the compromised user belongs to a group with a SID starting with S-1-5-80 then the specified target process uses the token of 'lsass'.

Table 1 – ShadowPad Modes

ANALYST NOTE: The shellcode is decrypted using a combination of bitwise XOR operations.

Configuration storage and structure

ShadowPad comes with an embedded encrypted configuration, which it locates by scanning its own shellcode (core module) with the following method (Python representation):

```
for dword in range( len(data) ):
  first_value = data[dword :dword+4]
  second_value = data[dword+4:dword+8]
  third_value = data[dword+8:dword+12]
  fourth_value = data[dword+12:dword+16]
  fifth_value = data[dword+16:dword+20]
  sixth_value = data[dword+20:dword+24]

  xor1 = int.from_bytes(second_value,'little') ^ 0x8C4832F1
  xor2 = int.from_bytes(fourth_value,'little') ^ 0xC3BF9669
  xor3 = int.from_bytes(sixth_value,'little') ^  0x9C2891BA

  if xor1 == int.from_bytes(first_value,'little') and xor2 ==
int.from_bytes(third_value,'little') and xor3 ==
int.from_bytes(fifth_value,'little'):
     print(f"found: {dword:02x}")
     encrypted = data[dword:]
     break
```

After locating it successfully, it starts searching in it for a specified byte that represents the type of data (e.g., 0x02 represents an embedded module). In total, we have identified the following types:

ID	Description
0x02	Embedded ShadowPad module.
0x80	ShadowPad configuration. It should start with the DWORD value 0x9C9D22EC.
0x90	XOR key used during the generation of unique names (e.g., registry key name)
0x91	DLL loader file data.
0x92	DLL loader file to load. File might have random appended data (Depends on the config's flag at offset 0x326).
0xA0	Loader's filepath

Table 2 – Shadowpad Data Types

Once one of the above bytes are located, ShadowPad reads the data (size is defined before the byte identifier) and appends the last DWORD value to the hardcoded byte array '1A9115B2D21384C6DA3C21FCCA5201A4'. Then it hashes (MD5) the constructed byte array and derives an AES-CBC 128bits key and decrypts the data.

In addition, ShadowPad stores, in an encrypted format, the following data in the registry with the registry key name being unique (based on volume serial number of C:\) for each compromised host:

1. ShadowPad configuration (0x80) data.
2. Proxy configuration. Includes proxy information that ShadowPad requires. These are the network communication protocol, domain/IP proxy and the proxy port.
3. Downloaded modules.

ShadowPad Network Servers

ShadowPad starts two TCP/UDP servers at 0.0.0.0. The port(s) is/are specified in the ShadowPad configuration. These servers work as a proxy between other compromised hosts in the network.

In addition, ShadowPads starts a raw socket server, which receives data and does one of the following tasks (depending on the received data):

1. Updates and sets proxy configuration to SOCKS4 mode.
2. Updates and sets proxy configuration to SOCKS5 mode.
3. Updates and sets proxy configuration to HTTP mode.

Network Communication

ShadowPad supports a variety of network protocols (supported by dedicated modules). For all of them, ShadowPad uses the same procedure to store and encrypt network data. The procedure's steps are:

1. Compress the network data using the QuickLZ library module.
2. Generates a random DWORD value, which is appended to the byte array '1A9115B2D21384C6DA3C21FCCA5201A4'. Then, the constructed byte array is hashed (MD5) and an AES-CBC 128bits key is derived (CryptDeriveKey).
3. The data is then encrypted using the generated AES key. In addition, Shadowpad encrypts the following data fields using bitwise XOR operations:

i. Command/Module ID: Command/Module ID ^ (0x1FFFFF * Hashing_Key – 0x2C7BEECE)
ii. Data_Size: Data_Size ^ (0x1FFFFFF * 0x7FFFFF * (0x1FFFFF * Hashing_Key – 0x2C7BEECE) – 0x536C9757 – 0x7C06303F)
iii. Command_Execution_State: Command_Execution_State ^ 0x7FFFFF * (0x1FFFFF * Hashing_Key – 0x2C7BEECE) – 0x536C9757

As a last step, ShadowPad encapsulates the above generated data into the following structure:

```
struct Network_Packet
{
 DWORD Hashing_Key;
 DWORD Command_ID_Module_ID;
 DWORD Command_Execution_State; //Usually contains any error codes.
 DWORD Data_Size;
 byte data[Data_Size];
};
```

If any server responds, it should have the same format as above.

Network Commands and Modules

During our analysis, we managed to extract a variety of ShadowPad modules with most of them having their own set of network commands. The table below summarises the identified commands of the modules, which we managed to recover.

Module	Command ID	Description
Main module	0xC49D0031	First command sent to the C2 if the commands fetcher function does not run in a dedicated thread.
Main module	0xC49D0032	First command sent to the C2 if the commands fetcher function does run in a dedicated thread.
Main module	0xC49D0033	Fingerprints the compromised host and sends the information to the C2.
Main module	0xC49D0032	(Received) Executes the network command fetcher

		function in a thread.
Main module	0xC49D0034	Sents an empty reply to the C2.
Main module	0xC49D0037	Echoes the server’s reply.
Main module	0xC49D0039	Sends number of times the Shadowpad files were detected to be deleted.
Main module	0xC49D0016	Deletes Shadowpad registry keys.
Main module	0xC49D0035	Enters sleep mode for 3 seconds in total.
Main module	0xC49D0036	Enters sleep mode for 5 seconds in total.
Main module	0xC49D0010	Retrieves Shadowpad execution information.
Main module	0xC49D0012	Updates Shadowpad configuration (in registry).
Main module	0xC49D0014	Deletes Shadowpad module from registry.
Main module	0xC49D0015	Unloads a Shadowpad module.
Main module	0xC49D0020	Retrieves Shadowpad current configuration (from registry).
Main module	0xC49D0021	Updates the Shadowpad configuration in registry and (re)starts the TCP/UDP servers.
Main module	0xC49D0022	Deletes Shadowpad registry entries and starts the TCP/UDP servers.
Main module	0xC49D0050	Retrieves Shadowpad proxy configuration from registry.
Main module	0xC49D0051	Updates Shadowpad proxy configuration.
Main module	0xC49D0052	Updates Shadowpad proxy configuration by index.
Main module	0xC49D0053	Sets Shadowpad proxy configuration bytes to 0
Main module	Any Module ID	Loads and initialises the specified module ID.
Files manager module	0x67520006	File operations (copy,delete,move,rename).
Files manager module	0x67520007	Executes a file.
Files manager module	0x67520008	Uploads/Downloads file to/from C2.
Files manager module	0x6752000A	Searches for a specified file.
Files manager module	0x6752000C	Downloads a file from a specified URL.
Files manager module	0x67520005	Timestomp a file.
Files manager module	0x67520000	Get logical drives information.
Files manager module	0x67520001	Searches recursively for a file.
Files manager module	0x67520002	Checks if file/directory is writable.
Files manager module	0x67520003	Creates a directory.
Files manager module	0x67520004	Gets files list in a given directory
TCP/UDP module	0x54BD0000	Loads TCP module and proxy data via it.
TCP/UDP module	0x54BD0001	Proxies UDP network data.
Desktop module	0x62D50000	Enumerates monitors.
Desktop module	0x62D50001	Takes desktop screenshot.
Desktop module	0x62D50002	Captures monitor screen.
Desktop module	0x62D50010	Gets desktop module local database file path.
Desktop module	0x62D50011	Reads and sends the contents of local database file to the C2.
Desktop module	0x62D50012	Writes to local database file and starts a thread that constantly takes desktop screenshots.
Processes manager module	0x70D0000	Gets processes list along with their information
Processes manager module	0x70D0001	Terminates a specified process
Network Connections module	0x6D0000	Gets TCP network table.
Network	0x6D0001	Gets UDP network table.

Connections module		
PIPEs module	0x23220000	Reads/Writes data to PIPEs.
Propagation module	0x2C120010	Get module's configuration.
Propagation module	0x2C120011	Transfer network data between C2 and PIPEs.
Propagation module	0x2C120012	Constant transfer of network data between C2 and PIPEs.
Propagation module	0x2C120013	Transfer network data between C2 and PIPEs.
Propagation module	0x2C120014	Constant transfer of network data between C2 and PIPEs.
Propagation module	0x2C120015	Transfer network data between C2 and PIPEs.
Propagation module	0x2C120016	Constant transfer of network data between C2 and PIPEs.
Propagation module	0x2C120017	Transfer network data between C2 and PIPEs.
Propagation module	0x2C120018	Transfer network data between C2 and PIPEs.
Scheduled tasks module	0x71CD0000	Gets a list of the scheduled tasks.
Scheduled tasks module	0x71CD0001	Gets information of a specified scheduled task.
Wi-Fi stealer module	0xDC320000	Collects credentials/information of available Wi-Fi devices.
Network discovery module	0xF36A0000	Collects MAC addresses.
Network discovery module	0xF36A0001	Collects IP addresses information.
Network discovery module	0xF36A0003	Port scanning.
Console module	0x329A0000	Starts a console mode in the compromised host.
Keylogger module	0x63CA0000	Reads the keylogger file and sends its content to the C2.
Keylogger module	0x63CA0001	Deletes keylogger file.

Table 3 – Modules Network Commands

Below are listed the available modules, which do not have network commands (Table 3).

Module ID	Description
E8B5	QUICKLZ library module.
7D82	Sockets connection module (supports SOCKS4, SOCKS5 and HTTP).
C7BA	TCP module.

Table 4 – Available modules without network commands

Below are listed the modules that we identified after analysing the main module of ShadowPad but were not recovered.

Module ID	Description
0x25B2	UDP network module.
0x1FE2	HTTP network module.
0x9C8A	HTTPS network module.
0x92CA	ICMP network module
0x64EA	Unknown

Table 5 – Non-Recovered ShadowPad Modules

Misc

1. ShadowPad uses a checksum method to compare certain values (e.g., if it runs under certain access rights). This method has been implemented below in Python:

```
ror = lambda val, r_bits, max_bits: \
((val & (2**max_bits-1)) >> r_bits%max_bits) | \
(val << (max_bits-(r_bits%max_bits)) & (2**max_bits-1))
rounds = 0x80
```

```
data = b""
output = 0xB69F4F21
max_bits = 32
counter = 0

for i in range( len(data) ):
 data_character = data[counter]
 if (data_character - 97)&0xff <= 0x19:
  data_character &= ~0x20&0xfffffff
  counter +=1
  output = (data_character + ror(output, 8,32)) ^ 0xF90393D1
  print ( hex( output ))
```

- Under certain modes, ShadowPad chooses to download and inject a payload from its command-and-control server. ShadowPad parses its command-and-control server domain/IP address and sends a HTTP request. The reply is expected to be a payload, which ShadowPad injects into another process.

ANALYST NOTE: In case the IP address/Domain includes the character '@', ShadowPad decrypts it with a custom algorithm.

Indicators of Compromise

IOC	Indicator Type	D
C:\wso2is-4.6.0\BVRPDiag.exe	File Path	Le ex si Po
C:\wso2is-4.6.0\BVRPDiag.tsi	File Path	
C:\wso2is-4.6.0\BVRPDiag.dll	File Path	Po
C:\wso2is-4.6.0\ModemMOH.dll	File Path	
C:\Windows\System32\spool\drivers\color\K7AVWScn.dll	File Path	Pr ur C:
C:\Windows\System32\spool\drivers\color\K7AVWScn.doc	File Path	U th lo Po
C:\Windows\System32\spool\drivers\color\K7AVWScn.exe	File Path	Le ex si Po
C:\Windows\System32\spool\drivers\color\secur32.dll	File Path	Sl D
C:\Windows\System32\spool\drivers\color\secur32.dll.dat	File Path	Sl Er Co
C:\Windows\System32\spool\drivers\color\WindowsUpdate.exe	File Path	Le ex si Sl
C:\Windows\Temp\WinLog\secur32.dll	File Path	Sl D
C:\Windows\Temp\WinLog\secur32.dll.dat	File Path	Sl Er Co
C:\Windows\Temp\WinLog\WindowsEvents.exe	File Path	Le ex si Sl
C:\ProgramData\7z.dll	File Path	Ar
C:\ProgramData\7z.exe	File Path	Ar
C:\Users\Public\AdFind.exe	File Path	Re to

C:\Users\Public\nbtscan.exe	File Path	R to
C:\Users\Public\start.bat	File Path	U sc su st of
C:\Users\Public\t\64.exe	File Path	U ex su m
C:\Users\Public\t\7z.exe	File Path	A
C:\Users\public\t\browser.exe	File Path	U at ex
C:\Users\Public\t\nircmd.exe	File Path	N sn lin al sc ta di us
C:\users\public\t\test.bat	File Path	U at sc
C:\Users\Public\test.bat	File Path	U at sc
C:\Users\Public\test.exe	File Path	U at ex
C:\Users\Public\test\Active Directory\ntds.dit	File Path	St fo
C:\Users\Public\test\registry\SECURITY	File Path	St fo du
C:\Users\Public\test\registry\SYSTEM	File Path	St fo du
C:\Users\Public\WebBrowserPassView.exe	File Path	N re cr w
C:\Windows\debug\adprep\P.bat	File Path	U at sc
C:\Windows\system32\spool\drivers\affair.exe	File Path	U at ex
C:\Windows\System32\spool\drivers\color\SessionGopher.ps1	File Path	D se in re to
C:\windows\system32\spool\drivers\color\tt.bat	File Path	U at sc
C:\Windows\Temp\best.exe	File Path	Tr
ip445.ps1	File Name	U Pc sc to ne re
ip445.txt	File Name	Su ou ip
nbtscan.exe	File Name	At
SOFTWARE: Classes\CLSID*\42BF3891	Registry Key	Er Sl

		cc
SOFTWARE: Classes\CLSID*\45E6A5BE	Registry Key	Er Sl cc
SOFTWARE: Classes\CLSID*\840EE6F6	Registry Key	Er Sl cc
SOFTWARE: Classes\CLSID*\9003BDD0	Registry Key	Er Sl cc
Software:Classes\CLSID*\51E27247	Registry Key	Er Sl cc
Software\Microsoft**\009F24BCCEA54128C2344E03CEE577E12504DD569C8B48AB8B7EAD5249778643	Registry Key	Er Sl m
Software\Microsoft**\5F336A90564002BE360DF63106AA7A7568829C6C084E793D6DC93A896C476204	Registry Key	Er Sl m
Software\Microsoft**\FF98EFB4C7680726BF336CEC477777BB3BEB73C7BAA1A5A574C39E7F4E804585	Registry Key	Er Sl m
D1D0E39004FA8138E2F2C4157FA3B44B	MD5 Hash	P
54B419C2CAC1A08605936E016D460697	MD5 Hash	U ba
B426C17B99F282C13593954568D86863	MD5 Hash	U ba re
7504DEA93DB3B8417F16145E8272BA08	MD5 Hash	Sl D
D99B22020490ECC6F0237EFB2C3DEF27	MD5 Hash	Sl D
1E6E936A0A862F18895BC7DD6F607EB4	MD5 Hash	Sl D
A6A19804248E9CC5D7DE5AEA86590C63	MD5 Hash	Sl D
4BFE4975CEAA15ED0031941A390FAB55	MD5 Hash	Sl D
87F9D1DE3E549469F918778BD637666D	MD5 Hash	Sl D
8E9F8E8AB0BEF7838F2A5164CF7737E4	MD5 Hash	Sl D

Mitre Att&ck

Tactic	Technique	ID	Description
Initial Access	Exploit Public-Facing Applications	T1190	Initial access was gained via the threat actor exploiting CVE-2022-29464 to create a web shell
Execution	Command and Scripting Interpreter: PowerShell	T1059:001	PowerShell based tools PowerView and SessionGopher were executed across the estate for reconnaissance and credential harvesting. Additionally, hands on keyboard commands were identified as being executed to confirm which version of the malware was present.
Execution	Command and Scripting Interpreter: Windows Command Shell	T1059:003	A scheduled task used by the threat actor was used to launch a Windows Command Shell. The purpose is not known.
Execution	Command and Scripting Interpreter: Python	T1059:006	Several compiled python binaries were identified. It is likely the binaries related to the creation of an FTP server.
Execution	Scheduled Task/Job: Scheduled Task	T1053	A scheduled task named “update” was observed and configured to execute a command prompt on multiple hosts throughout the environment. Upon successful execution of the task the threat actor then deleted the task from the host
Execution	Exploitation for Client	T1203	The threat actor leveraged CVE-2022-29464 to deploy web shells and allow remote command execution on

	Execution		patient zero.
Execution	Windows Management Instrumentation (WMI)	T1047	WMI was used by the threat actor to carry out reconnaissance activity.
Persistence	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	T1547.001	A run key for the local administrator was created to execute the malicious backdoor.
Persistence	Create or Modify System Process: Windows Service	T1543.003	Two malicious services were deployed widely across the estate for persistence of the backdoors. Both services execute a legitimate binary which is stored in the same location as a malicious DLL, when executed the legitimate binary would side load the malicious DLL containing the backdoor.
Privilege Escalation	Valid Accounts: Domain Accounts	T1078.002	The threat actor was primarily using domain administrator credentials to move laterally throughout the attack, allowing them to blend in with legitimate administrator activity.
Defence Evasion	Impair Defenses: Downgrade Attack	T1562.010	The threat actor was observed utilising PowerShell downgrades, this is typically used by threat actors to avoid the script logging capabilities of PowerShell version 5+
Defence Evasion	Indicator Removal on Host: File Deletion	T1070.004	The threat actor routinely removed the majority of tooling deployed throughout the attack from hosts upon completion of their objectives.
Defence Evasion	Indicator Removal on Host: Timestomp	T1070.006	The threat actor timestomped all files relating to the backdoors including the legitimate binary and the malicious DLL.
Defence Evasion	Modify Registry	T1112	The modules for ShadowPad were stored within the registry in an encrypted format. The keys for the stored data are generated depending on the volume serial number of the host.
Defence Evasion	Obfuscated Files or Information	T1027	The ShadowPad configuration was stored within an encrypted registry hive. The keylogger module of ShadowPad created an encrypted output file on the host.
Defence Evasion	Masquerading: Rename System Utilities	T1036.003	The threat actor leveraged a legitimate Windows DLL, secur32.dll, as the name of the configuration file for the ShadowPad backdoor.
Defence Evasion	Process Injection: Process Hollowing	T1055.012	Upon execution ShadowPad spawns a sacrificial process, which then utilises the technique of process hollowing to inject into the process.
Defence Evasion	Hide Artefacts: Hidden Files and Directories	T1564.001	Several malicious files were identified as having the NTFS attribute of hidden.
Defence Evasion	Hijack Execution Flow: DLL Search Order Hijacking	T1574.001	The backdoors leveraged DLL Search Order Hijacking.
Credential Access	Credentials from Password Stores: Credentials from Web Browsers	T1555:003	The NirSoft tool WebBrowserPassView.exe was also identified as being executed by the attacker.
Credential Access	Credentials from Password Stores: Windows Credential Manager	T1555.004	Credential harvesting which indicated credentials from Windows Credential Manager were collected was identified on a domain controller.
Credential Access	OS Credential Dumping: LSASS Memory	T1003.001	ProcDump.exe was leveraged on patient zero during the attack in order to dump credentials stored in the process memory of Local Security Authority Subsystem Service (LSASS).
Credential Access	OS Credential Dumping: NTDS	T1003.003	The NTDS.dit was dumped and exfiltrated from a domain controller for each domain.
Credential	Unsecured	T1552.001	Several instances of passwords in plaintext files were

Access	Credentials: Credentials in Files		observed on hosts where ShadowPad was installed/
Credential Access	Input Capture: Keylogging	T1056:001	ShadowPad instances had a Keylogger module installed.
Discovery	File and Directory Discovery	T1083	Tree.exe was used to enumerate files and directories on compromised hosts.
Discovery	Network Share Discovery	T1135	A PowerShell script named ip445.ps1 was used throughout the attack to enumerate network shares across the Windows estate.
Discovery	System Network Configuration Discovery	T016	AdFind.exe can extract subnet information from Active Directory.
Discovery	Account Discovery: Domain Account	T1087.002	AdFind.exe can enumerate domain users.
Discovery	Domain Trust Discovery	T1482	AdFind.exe can gather information about organizational units (OUs) and domain trusts from Active Directory.
Discovery	Permission Groups Discovery: Domain Groups	T1069	AdFind.exe can enumerate domain groups.
Discovery	Remote System Discovery	T1018	AdFind.exe has the ability to query Active Directory for computers.
Lateral Movement	Remote Services: Remote Desktop Protocol	T1021.001	RDP was used by the threat actor to laterally move. It is unknown whether this was a deliberate act to move estates or if the threat actor was attempting to move to another domain.
Lateral Movement	Remote Services: SMB/Windows Admin Shares	T1021.002	The Powerview module of Powersploit was used to enumerate all SMB shares across the environment.
Lateral Movement	Remote Services: Windows Remote Management	T1021.006	WinRM was used by the actor during periods of network reconnaissance.
Lateral Movement	Remote Services: Distributed Component Object Model	T1021.003	Anti-virus alerts showed the threat actor as utilising WMI to laterally move to hosts across the network.
Collection	Automated Collection	T1119	Large scale credential harvesting was conducted against remote hosts from a domain controller.
Collection	Data Staged: Remote Data Staging	T1074.002	Credentials harvested by the threat actor were collected on a domain controller, prior to exfiltration.
Collection	Input Capture: Keylogging	T1056.001	ShadowPad instances had a Keylogger module installed which allowed them to capture the input of interactive sessions. The output was stored on disk in encrypted database files.
Collection	Archive Collected Data: Archive via Utility	T1560.001	The actor was routinely observed archiving collected data via 7zip.
Command and Control	Encrypted Channel	T1573	ShadowPad configurations indicated Command and Control communications were sent via port 443.
Command and Control	Proxy: Internal Proxy	T1090.001	ShadowPad instances had a Proxy module installed. It was identified that a proxy module was installed and was interacting via port 445.
Exfiltration	Exfiltration Over C2 Channel	T1041	ShadowPad has the capability to exfiltrate data.

---

[1] https://www.secureworks.com/research/shadowpad-malware-analysis

[2] https://www.pwc.co.uk/issues/cyber-security-services/research/chasing-shadows.html

[3] https://nvd.nist.gov/vuln/detail/CVE-2022-29464