

Spear Phishing Campaign Delivers Buer & Bazar | Zscaler Blog

By Mohd Sadique, Atinderpal Singh

Published: 2020-09-29 · Archived: 2026-04-05 16:56:43 UTC

Zscaler ThreatLabZ became aware of a prevalent phishing campaign targeting employees of various organizations. During the past couple of weeks, many enterprise users have been getting spear phishing emails indicating that their employment with the company has been terminated.

These emails contain a Google document link that leads to the Bazar backdoor (from the TrickBot gang). What's interesting is that this campaign also used the Buer loader, which is the first time we have seen these two malware strains used together.

Use of the Buer loader by the TrickBot gang comes as no surprise as this group is known to work with different malware groups. In the past, the TrickBot gang has also worked with other botnets, such as Emotet.

Campaign

In this email campaign, instead of relying on attachments, the attackers included links to what appeared to be a legitimate Google Docs document, which itself contained links to malicious files hosted on Google Drive or, in some cases, hosted elsewhere. In some previous phishing email campaigns, attackers leveraged SendGrid to distribute the initial emails to hide the Google Drive links in the documents behind a SendGrid URL as a way to bypass traditional defences.

Samples of emails that we have seen are shown in Figure 1 and Figure 2.

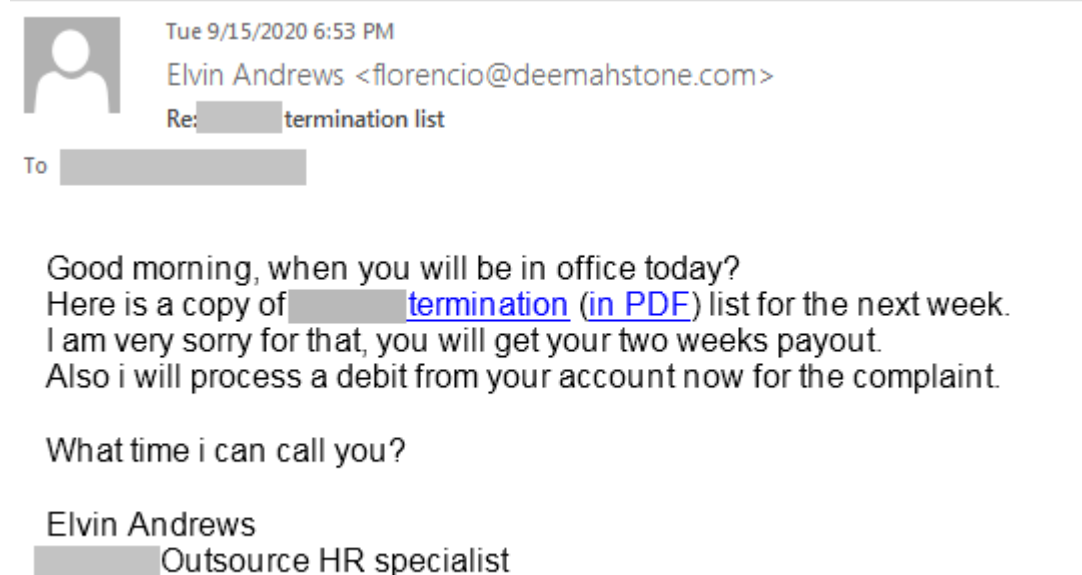


Figure 1: One of the spear phishing email templates targeting an employee.

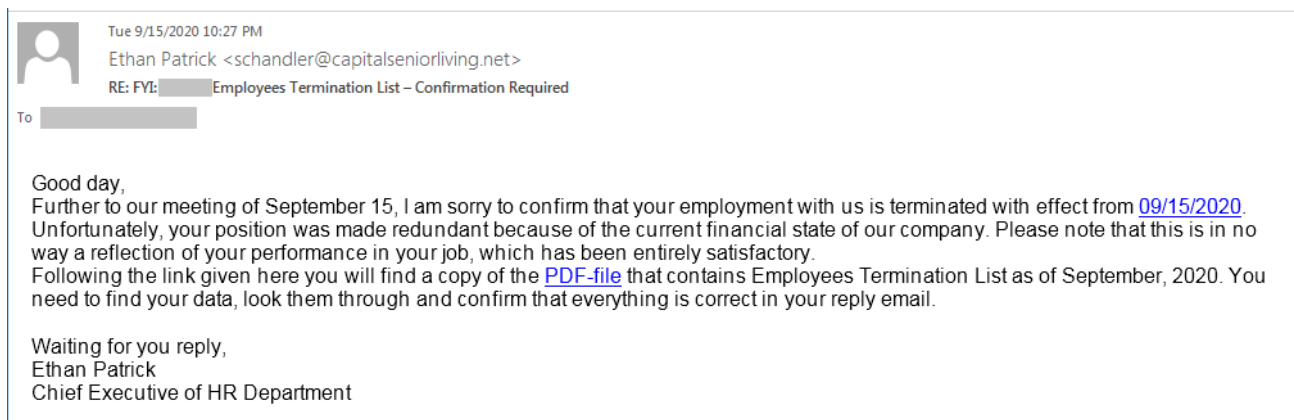


Figure 2: Another spear phishing email template

The link in both emails is a Google Docs link claiming to host a PDF file with a list of employees that have been terminated, as shown in Figure 3.

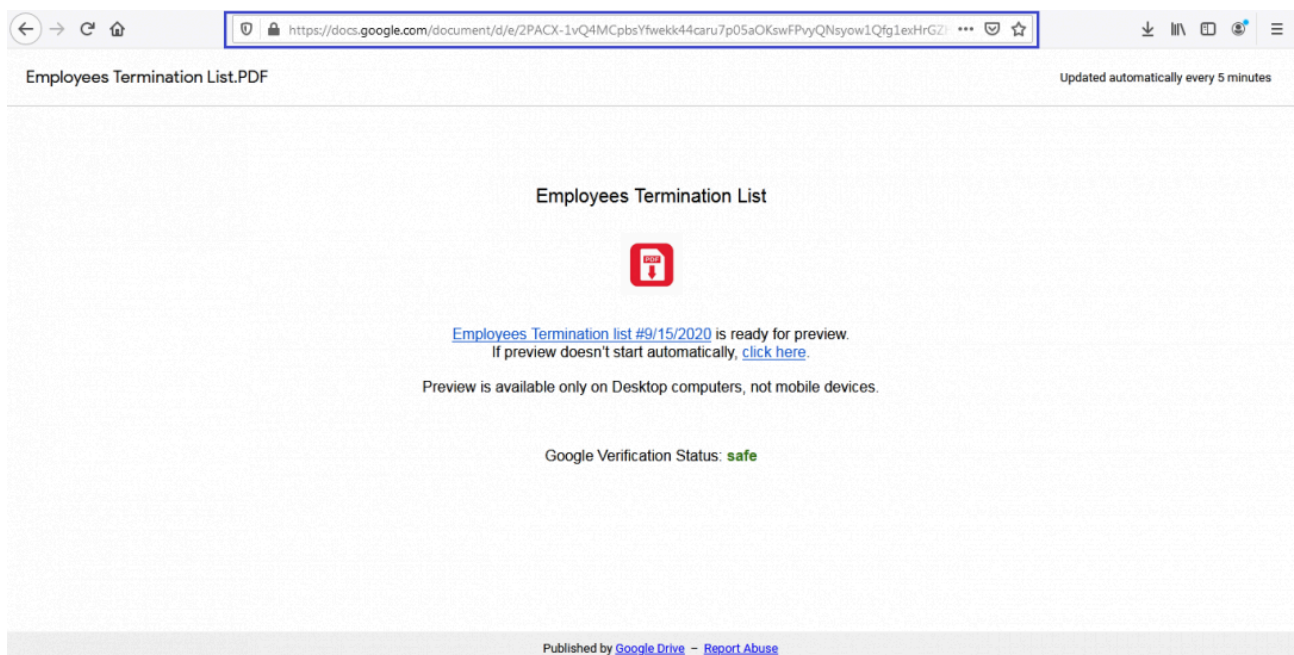


Figure 3: The link to the fake Google Doc containing the download link.

The link in the Google Doc redirects to the URL `unitedyfl[.]com/print_preview.exe` to download the malware payload.

Although, the use of target names with actuating themes is not new to this group, there has been a significant uptick in the number of emails received and this campaign has been persistently active for the past few weeks.

Packer

In most cases, the payload that is downloaded is the Bazar malware but, in some cases, it is the Buer loader. The packer used in both malware payloads is identical. Most notably, the packed binaries are exe files with a randomly

named export function. The export function is responsible for payload decryption and injection.

First, a shellcode is decrypted, which further decrypts a headerless PE loader that has the final payload in its overlay. The headerless loader allocates memory, maps the payload into memory with proper permissions, and finally transfers control to it. In this campaign, no process self-injection is used to load the payload.

Address	Hex	ASCII
001E0000	00 00 00 00
001E0010	00 00 00 00
001E0020	00 00 00 00
001E0030	00 00 00 00
001E0040	00 00 00 00
001E0050	00 00 00 00
001E0060	00 00 00 00
001E0070	00 00 00 00
001E0080	00 00 00 00
001E0090	00 00 00 00
001E00A0	00 00 00 00
001E00B0	00 00 00 00
001E00C0	00 00 00 00
001E00D0	43 2F 6A 5FPE.L...
001E00E0	08 01 0C 00c.....!
001E00F0	80 27 00 000.....
001E0100	00 10 00 000.....
001E0110	06 00 00 000.....
001E0120	00 00 00 00e.....
001E0130	00 00 10 000.....
001E0140	00 00 00 000.<.....
001E0150	00 00 00 00a.....
001E0160	00 00 00 00a.....
001E0170	00 00 00 00
001E0180	00 00 00 00
001E0190	00 00 00 00
001E01A0	00 30 00 000.\.....
001E01B0	00 00 00 00
001E01C0	2F 74 65 78text.....

Figure 4: The decrypted header less PE loader.

001E3F80	00 00 00 00
001E3F90	00 00 00 00
001E3FA0	00 00 00 00
001E3FB0	00 00 00 00
001E3FC0	00 00 00 00
001E3FD0	00 00 00 00
001E3FE0	00 00 00 00
001E3FF0	00 00 00 00
001E4000	01 00 00 00e.....
001E4010	10 00 00 000.....
001E4020	45 72 72 6F	Error protecting
001E4030	20 6D 65 6D	memory page....
001E4040	47 65 74 4E	GetNativeSystemI
001E4050	6E 66 6F 00	nfo.k.e.r.n.e.l
001E4060	33 00 32 00	3.2...d.l.l....
001E4070	4D 5A 90 00	MZ.....ÿ..
001E4080	B8 00 00 00e.....
001E4090	00 00 00 00
001E40A0	00 00 00 00
001E40B0	0E 1F BA 0E!..L!Th
001E40C0	69 73 20 70	is program canno
001E40D0	74 20 62 65	t be run in DOS
001E40E0	6D 6F 64 65	mode...\$.....
001E40F0	28 13 9E CB	(..Ëlrð.lrð.lrð.
001E4100	37 1A F1 99	7.ñ.drð.lrñ.%rð.
001E4110	AF 7D AD 98	}.orð.é.ù.Orð.
001E4120	E9 02 F2 99	é.ò.mrð.Richlrð.
001E4130	00 00 00 00
001E4140	50 45 00 00	PE.L...øh_....
001E4150	00 00 00 00a.....p..

Figure 5: The payload embedded at the end of the loader.

Bazar loader and Bazar backdoor

The Bazar backdoor is a new stealthy malware, part of the TrickBot group’s toolkit arsenal and leveraged for high-value targets. The Bazar loader is used to download and execute the Bazar backdoor on the target system. The goal of this backdoor is to execute binaries, scripts, modules, kill processes, and then remove itself from the compromised machine. The samples used in this campaign heavily rely on control flow obfuscation. The detailed analysis report about this backdoor can be found [here](#).

The Bazar loader downloads the Bazar backdoor from the C&C using the following URI format: {C&C}/api/v\d{3}

The downloaded payload is XOR-encrypted and can be decrypted using the script provided in the appendix.

The downloaded malware was successfully captured by the Zscaler Cloud Sandbox:

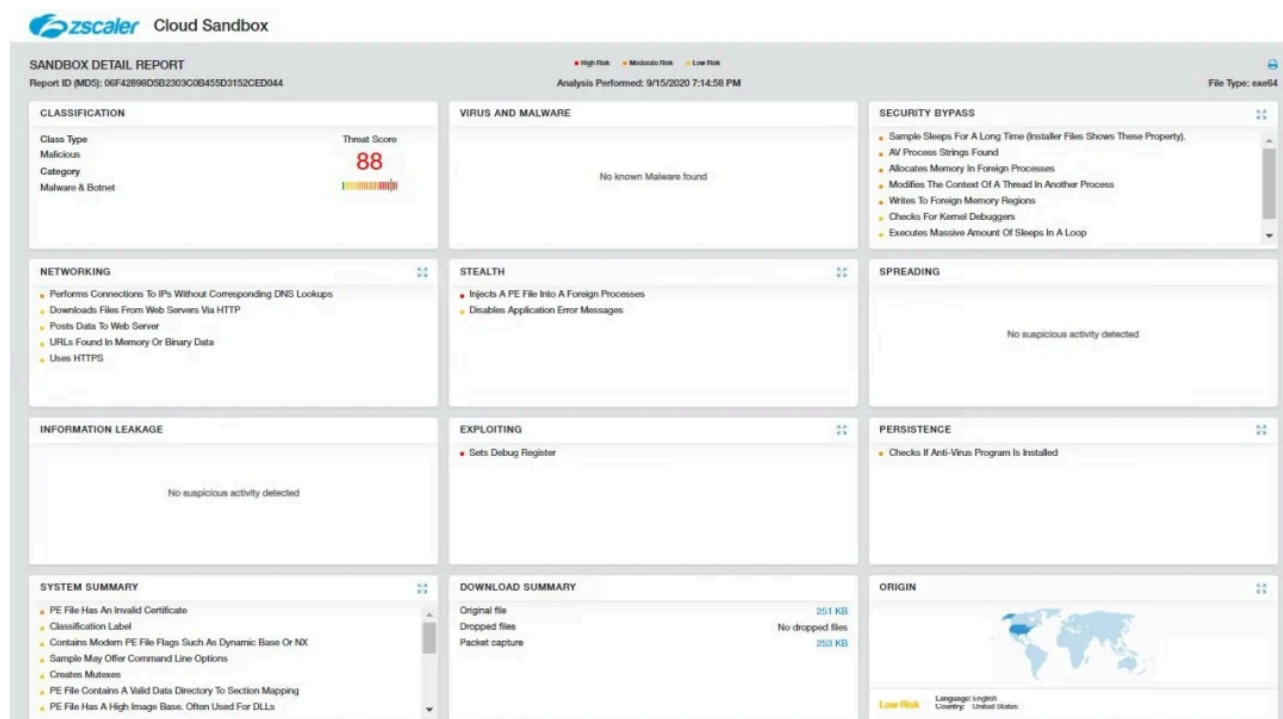


Figure 6: The Zscaler Cloud Sandbox report.

The C&C TLS communications of the Bazar backdoor have been using certificates created in the same manner

that TrickBot certificates have been created. The C&C server TLS certificate is shown in Figure 7.

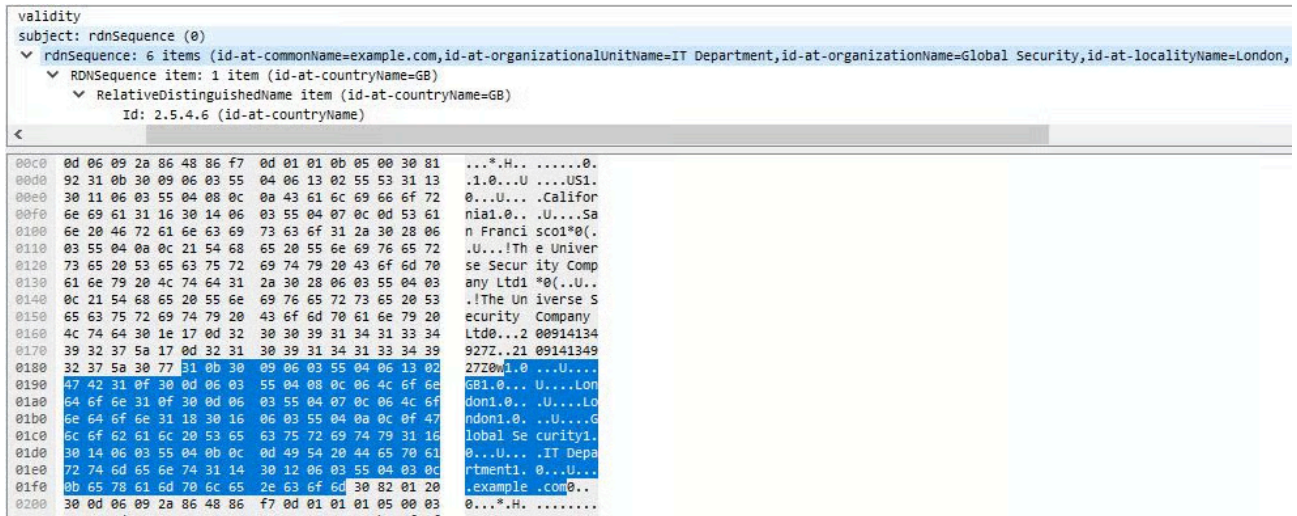


Figure 7: The Bazar/TrickBot TLS certificate.

Researchers also observed that the backdoor downloads and executes the Cobalt Strike pentesting and post-exploitation toolkit on the victim's machine within some period of time after the infection. By deploying Cobalt Strike, it is clear that this stealthy backdoor is being used to gain a foothold in corporate networks so that ransomware can be deployed, data can be stolen, or network access could be sold to other threat actors.

Buer loader

The Buer loader was first discovered around the end of 2019. It is a very capable malware written in C and primarily sold on Russian underground forums for around US\$400. Notably, this malware does not function in the CIS. It has most of the important strings encrypted and APIs are loaded by hash, just like most of the sophisticated malware these days. We are not going to go into technical details because detailed analysis of this has already [published](#).

The Buer loader was captured by the Zscaler Cloud Sandbox.

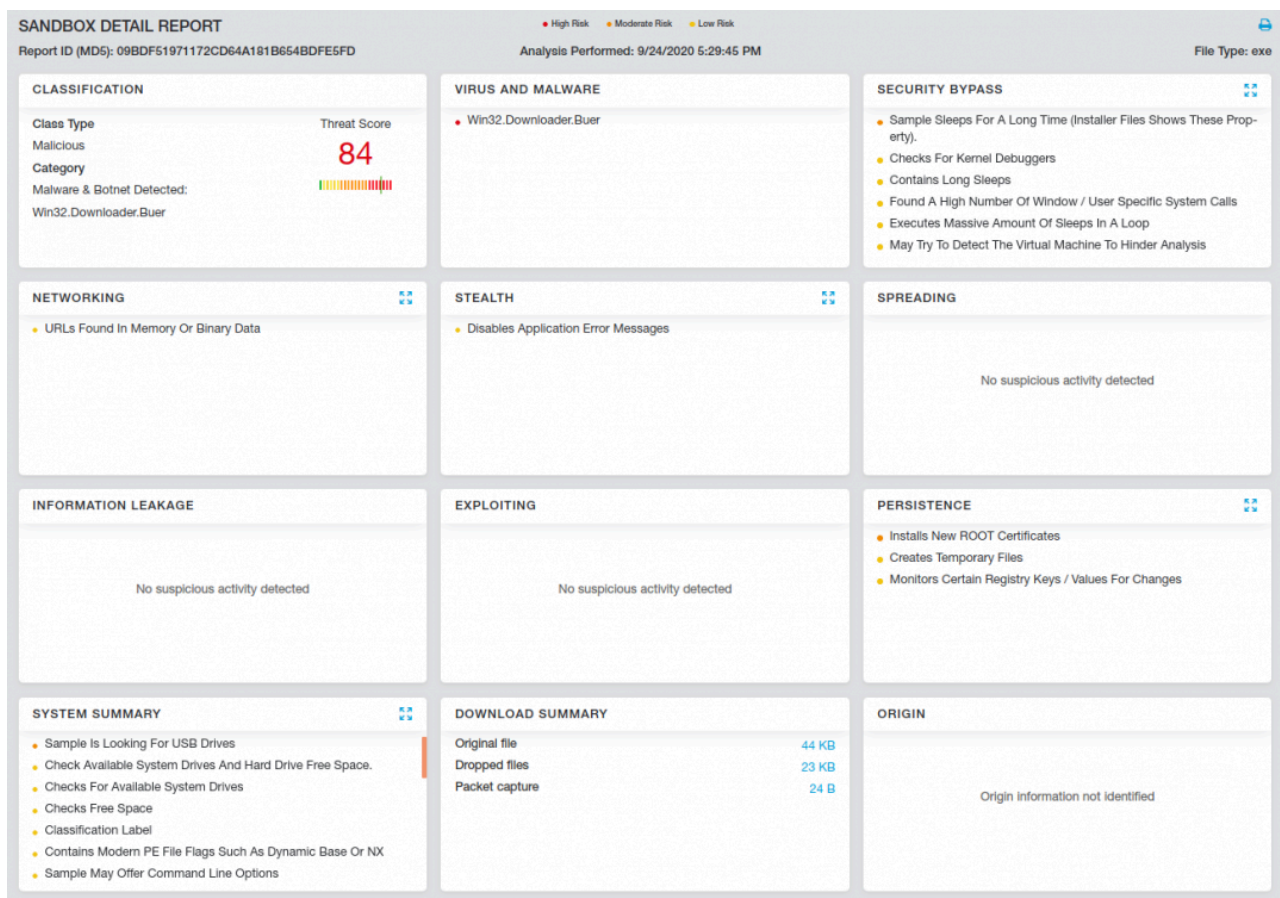


Figure 8: The Zscaler Cloud Sandbox report for the Buer loader.

In addition to sandbox detections, Zscaler’s multilayered cloud security platform detects indicators at various levels:

- [Win32.Trojan.Buerloader](#)
- [Win32.Backdoor.Bazar](#)

Conclusion

The TrickBot group has been running similarly themed campaigns for some time. The targeted nature of the campaign with subject lines having the organization’s name makes these campaign’s highly effective compared to generic spray-and-pray attacks. But even these specially crafted attacks are not immune from a pair of vigilant eyes and the right set of tools. We at Zscaler ThreatLabZ are always on the lookout for bad stuff—be it for our company or for our customers—to provide protection against it.

Last but not the least, always be attentive while opening any email links or attachments. Even if there is a tiny bit of suspicion, verify the email or get it reviewed thoroughly by your security team before proceeding further.

IOCs

MD5

Fa0322fb70610d6e67585588184eda39 (Buer loader)

06f42898d5b2303c0b455d3152ced044 (Bazar loader)

04a20c9f33023439b612935b6901917f (Bazar loader)

951acc18e4f14471f49235327e0c1ccc (Bazar loader)

4bb9a709958a1790a6bc257a9b5cb48e (Bazar loader)

03e699324d06bd3d597994f5df893048 (Bazar backdoor group: t1)

Distribution and document URLs

[http://unitedyfl\[.\]com/print_preview.exe](http://unitedyfl[.]com/print_preview.exe)

[https://docs.google\[.\]com/document/u/1/d/e/2PACX-1vTwnIt9tXcgRxaOME9G3yErRp50dGxW1EKoTeIAYZwkMEg4j8fOpU9kP7xMJ6pufKfzsoETJwX5ZMM5/pub](https://docs.google[.]com/document/u/1/d/e/2PACX-1vTwnIt9tXcgRxaOME9G3yErRp50dGxW1EKoTeIAYZwkMEg4j8fOpU9kP7xMJ6pufKfzsoETJwX5ZMM5/pub)

[https://docs.google\[.\]com/document/u/1/d/e/2PACX-1vSE2BfEV4tOmHOpMzeBhWbyajWwjxajBvm1YpJSRWyDL-qXbnSsu-OHhyuT2Y4mbZ72uPT9uToZWvo2/pub](https://docs.google[.]com/document/u/1/d/e/2PACX-1vSE2BfEV4tOmHOpMzeBhWbyajWwjxajBvm1YpJSRWyDL-qXbnSsu-OHhyuT2Y4mbZ72uPT9uToZWvo2/pub)

[https://docs.google\[.\]com/document/u/1/d/e/2PACX-1vTCf1OgjnHoahnZ0BMwCFRU62HyC85BfeiX7NGPiwvrqr8P--Y_5Mab9wAJjCIcldWv8wvKvXfuiK/pub](https://docs.google[.]com/document/u/1/d/e/2PACX-1vTCf1OgjnHoahnZ0BMwCFRU62HyC85BfeiX7NGPiwvrqr8P--Y_5Mab9wAJjCIcldWv8wvKvXfuiK/pub)

[https://docs.google\[.\]com/document/d/e/2PACX-1vQ4MCpbsYfwekk44caru7p05aOKswFPvyQNsow1Qfg1exHrGZHaqOmWcnSeAxmDK2V1i3ml9DP8kYT/pub](https://docs.google[.]com/document/d/e/2PACX-1vQ4MCpbsYfwekk44caru7p05aOKswFPvyQNsow1Qfg1exHrGZHaqOmWcnSeAxmDK2V1i3ml9DP8kYT/pub)

[https://docs.google\[.\]com/document/d/e/2PACX-1vRl0GvrO4JO8Rs4v1BTtXmsMThv1M413Z14onQl-TkrsXZEOR1zF8gKu3GDOwFBN0kaw5g7oC7lIE/pub](https://docs.google[.]com/document/d/e/2PACX-1vRl0GvrO4JO8Rs4v1BTtXmsMThv1M413Z14onQl-TkrsXZEOR1zF8gKu3GDOwFBN0kaw5g7oC7lIE/pub)

[https://docs.google\[.\]com/document/d/e/2PACX-1vR0NwqguWEFX4ZilvsxKSaJQbUfXpfK5fvWxbxUBJfPzmvGuxHS7bltp9cjpJ0RrvrdlAxeKpSjDKQ/pub](https://docs.google[.]com/document/d/e/2PACX-1vR0NwqguWEFX4ZilvsxKSaJQbUfXpfK5fvWxbxUBJfPzmvGuxHS7bltp9cjpJ0RrvrdlAxeKpSjDKQ/pub)

C&C

Buer loader

104.248.83[.]113

Bazar loader

164.68.107[.]165

91.235.129[.]164

37.220.6[.]126

195.123.241[.]194

82.146.37[.]128
85.143.221[.]85
164.132.76[.]76
54.37.237[.]253

Some of the URIs seen in this campaign include

- /api/v190 - Download Updated Bazar loader(64 - bit)
- /api/v192 - Download Bazar backdoor(64 - bit)
- /api/v202 - (Server did not respond with payload at the time of analysis)
- /api/v207 - (Server did not respond with payload at the time of analysis)

PDB string

c:\Users\Mr.Anderson\Documents\Visual Studio 2008\Projects\Anderson\x64\Release\Anderson.pdb

Some of the subject lines observed

Re: {Target Company Name} termination list
Re: {Target Company Name} avoiding
FW: Urgent: {Target Company Name}: A Customer Complaint Request – Prompt Action Required
RE: FYI: {Target Company Name} Employees Termination List – Confirmation Required
Re: complaint request
Re: my call, {Target Company Name}.
Re: {Target Company Name} - my visit
Re: can't call you

MITRE ATT&CK

ID	Technique
T1566.002	Phishing: Spearphishing Link
T1566.003	Phishing: Spearphishing via Service
T1204.001	User Execution: Malicious Link
T1204.002	User Execution: Malicious File
T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder
T1055.013	Process Injection: Process Doppelgänger
T1055.012	Process Injection: Process Hollowing
T1027.002	Obfuscated Files or Information: Software Packing
T1140	Deobfuscate/Decode Files or Information
T1036.005	Masquerading: Match Legitimate Name or Location
T1087	Account Discovery
T1010	Application Window Discovery
T1083	File and Directory Discovery
T1057	Process Discovery

T1012	Query Registry
T1018	Remote System Discovery
T1082	System Information Discovery
T1033	System Owner/User Discovery
T1124	System Time Discovery
T1119	Automated Collection
T1005	Data from Local System
T1053.002	Scheduled Task/Job: At (Windows)
T1547.004	Boot or Logon Autostart Execution: Winlogon Helper DLL
T1071.001	Application Layer Protocol: Web Protocols
T1568.002	Dynamic Resolution: Domain Generation Algorithms
T1020	Automated Exfiltration
T1041	Exfiltration Over C2 Channel
T1568.002	Dynamic Resolution: Domain Generation Algorithms

Appendix

Script to decrypt downloaded Bazar backdoor

```
key = "20200915"
data = open("v190", 'rb').read()
out = ""
for i in range(len(data)):
    out += chr(ord(data[i]) ^ ord(key[i%len(key)]))
of = open('dec1', 'wb')
of.write(out)
of.close()
#Note: Key can vary between downloader samples
```

Buer strings

```
Uc3nakqfdpncFjc
powershell.exe -Command "& {Add-MpPreference -ExclusionPath
update
Kdc23icmQoc21f
open
.dll
rundll32
regsvr32
powershell.exe "-Command" "if((Get-ExecutionPolicy ) -ne 'AllSigned') { Set-ExecutionPolicy -Scope
%02x
POST
Content-Type: application/x-www-form-urlencoded
runas
%s, "%s"
Software\Microsoft\Windows\CurrentVersion\RunOnce
{%s-%d-%d}
ntdll.dll
myyux?44659379=3=83684
myyux?44659379=3=83684
myyux?44659379=3=83684
myyux?44659379=3=83684
myyux?44659379=3=83684
UndefinedTypeError>>1I5480%C9#5=0=B8
hd00kaN3/Iqc7_Kdh
secinit.exe
false
true
null
https://104.248.83.13/
api/update/
https://104.248.83.13/
```

```
api/update/  
X40Ivc07uWS  
update  
statusCode  
AccessToken  
method  
x64  
exeLocal  
memload  
memloadex  
api/download/  
api/downloadmodule/  
download_and_exec  
download_and_exec  
regsrv32  
rundll  
rundllx  
parameters  
autorun  
explorer.exe  
api/module/  
modules  
loadDllmem  
Admin  
User  
Windows 10  
Windows Server 2019/Server 2016  
Windows 8.1  
Windows Server 2012 R2  
Windows 8  
Windows Server 2012  
Windows 7  
Windows Server 2008 R2  
Windows XP  
SQCPJICW  
X40Ivc07uWS  
Unknown  
x32  
x64  
LdrLoadDll  
RtlCreateUserThread  
LdrGetProcedureAddress  
RtlFreeUnicodeString  
RtlAnsiStringToUnicodeString  
RtlInitAnsiString  
Mozilla/5.0 (Apple-iPhone7C2/1202.466; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko  
X40Ivc07uWS
```

```
dllhost.exe
dllhost.exe
Software\Microsoft\Windows NT\CurrentVersion\Winlogon
Shell
open
akb,cvc
%ALLUSERSPROFILE%
Ostersin
\AutoReg.exe
" ensgJJ
ensgJJ
explorer.exe
secinit.exe
shell32.dll
Winhttp.dll
advapi32.dll
user32.dll
netapi32.dll
NtWriteVirtualMemory
Lr?jjma_rcTgprs_jKckmpw
JbpEcrNpmacbspc?bbpcqq
Lr0scpwTgprs_jKckmpw
LrDpccTgprs_jKckmpw
LrNpmrcarTgprs_jKckmpw
LrPc_bTgprs_jKckmpw
LrEcrAmlrcvrRfpc_b
LrQcrAmlrcvrRfpc_b
```

Buer loader API hashes and corresponding API names

```
0x69f7df2a -> advapi32_GetTokenInformation
0xe79d18d6 -> kernel32_OpenProcessToken
0x47979a8f -> advapi32_GetCurrentHwProfileW
0x19e1e0c2 -> kernel32_RegCreateKeyExW
0xd45f73b5 -> kernel32_RegCloseKey
0xcb5998e2 -> kernel32_RegSetValueExW
0xce636ff5 -> advapi32_GetSidSubAuthority
0xaf7f658e -> winhttp_WinHttpOpen
0x20b4c051 -> winhttp_WinHttpSetTimeouts
0x8ef04f02 -> winhttp_WinHttpCrackUrl
0x9f47a05e -> winhttp_WinHttpConnect
0x1dd1d38d -> winhttp_WinHttpOpenRequest
0x26d17a4e -> winhttp_WinHttpSendRequest
0xb20e6a35 -> winhttp_WinHttpGetIEProxyConfigForCurrentUser
0x1ef97964 -> winhttp_WinHttpGetProxyForUrl
```

```
0x8678c3f6 -> winhttp_WinHttpSetOption
0xea74138b -> winhttp_WinHttpWriteData
0x80cc5bd7 -> winhttp_WinHttpReadData
0x6c3f3920 -> winhttp_WinHttpReceiveResponse
0xde67ac3c -> winhttp_WinHttpQueryHeaders
0x710832cd -> winhttp_WinHttpQueryDataAvailable
0x9964b3dc -> winhttp_WinHttpCloseHandle
0x302ebe1c -> kernel32_VirtualAlloc
0x4247bc72 -> kernel32_VirtualQuery
0x1803b7e3 -> kernel32_VirtualProtect
0x1a4b89aa -> kernel32_GetCurrentProcess
0x8a8b4676 -> kernel32_LoadLibraryA
0x1acae7a -> kernel32_GetProcAddress
0x61eebd02 -> kernel32_GetModuleHandleW
0x8a8b468c -> kernel32_LoadLibraryW
0xab489125 -> kernel32_GetNativeSystemInfo
0x34590d2e -> kernel32_GetLastError
0x5b3716c6 -> kernel32_GlobalFree
0xe183277b -> kernel32_VirtualFree
0x62f1df50 -> kernel32_VirtualFreeEx
0xdd78764 -> kernel32_VirtualAllocEx
0xf3cf5f6f -> kernel32_GetModuleFileNameW
0xae7a8bda -> kernel32_CloseHandle
0x29e91ba6 -> kernel32_HeapSize
0xe3802c0b -> kernel32_HeapAlloc
0x864bde7e -> kernel32_GetProcessHeap
0x12dfcc4e -> kernel32_ExitProcess
0x7722b4b -> kernel32_TerminateProcess
0xb4f0f46f -> kernel32_CreateProcessW
0xff5ec2ce -> kernel32_ExitThread
0x4b3e6161 -> kernel32_TerminateThread
0xed619452 -> kernel32_CreateMutexW
0x7bffe25e -> kernel32_OpenMutexW
0xf785ce6 -> kernel32_ReadFile
0xe6886cef -> kernel32_WriteFile
0x1a7f0bab -> kernel32_CreateFileW
0xbd9a937d -> kernel32_GetFileSize
0x617ea42b -> kernel32_DeleteFileW
0x6659de75 -> kernel32_WriteProcessMemory
0xc56e656d -> kernel32_GetCommandLineW
0x78c1ba50 -> kernel32_ExpandEnvironmentStringsW
0x2e0ccb63 -> kernel32_CreateDirectoryW
0x5c62ca81 -> kernel32_WaitForSingleObject
0x8edf8b90 -> kernel32_OpenProcess
0x8a62152f -> kernel32_CreateToolhelp32Snapshot
0xc9112e01 -> kernel32_Process32NextW
0x63f6889c -> kernel32_Process32FirstW
```

```
0x4b9358fc -> kernel32_DuplicateHandle
0x24e2968d -> kernel32_GetComputerNameW
0x110e739a -> kernel32_GetVolumeInformationW
0xf7643b99 -> kernel32_GetThreadContext
0x3cc73360 -> kernel32_ResumeThread
0x77643b9b -> kernel32_SetThreadContext
0x1c2c653b -> ntdll_memset
0x1c846140 -> ntdll_memcpy
0x932d8a1a -> ntdll_NtDelayExecution
0x9716d04e -> ntdll_NtReleaseMutant
0x6f7f7a64 -> ntdll_RtlGetVersion
0x996cc394 -> ntdll_ZwUnmapViewOfSection
0xabf93436 -> ntdll_strtol
0x2bd04fd1 -> ntdll_iswctype
0x26a5553c -> ntdll_strstr
0x4117fd0e -> ntdll_NtQueryDefaultLocale
0xd24c9118 -> ntdll_RtlCreateUserThread
0xd52ff865 -> ntdll_NtQueryVirtualMemory
0x339c09fb -> ntdll_NtQueryInformationProcess
0x6a13016e -> ntdll_NtSetInformationThread
0x6debaaa9 -> ntdll_NtFilterToken
0xd584ba6c -> shell32_SHGetFolderPathW
0x375eadf4 -> shell32_CommandLineToArgvW
0xba1eb35b -> shell32_ShellExecuteW
0xf674afe0 -> user32_wsprintfW
```

References

<https://www.bleepingcomputer.com/news/security/bazarbackdoor-trickbot-gang-s-new-stealthy-network-hacking-malware/>

<https://www.vkremez.com/2020/04/lets-learn-trickbot-bazarbackdoor.html>

<https://krabsonsecurity.com/2019/12/05/buer-loader-new-russian-loader-on-the-market-with-interesting-persistence/>

Explore more Zscaler blogs

Source: <https://www.zscaler.com/blogs/research/spear-phishing-campaign-delivers-buer-and-bazar-malware>