

Adversaries exploit Ivanti's zero-day vulnerabilities

Original report published on: Feb 29, 2024^[1]

Executive Summary

On February 29, 2024, Cybersecurity and Infrastructure Security Agency (CISA) released an advisory that threat actors are exploiting previously identified vulnerabilities in Ivanti Connect Secure and Ivanti Policy Secure gateways, including CVE-2023-46805 (CVSS 8.2), CVE-2024-21887 (CVSS 9.1) and CVE-2024-21893 (CVSS 8.1).

For organisations that rely on Ivanti for secure remote access, these vulnerabilities could disrupt services and compromise sensitive data if exploited.

Background

Ivanti's released an Integrity Checker Tool (ICT) to detect vulnerable versions. CISA identified that the previous version of internal and external ICT can run to detect vulnerable versions but not previous compromise by rootkit as adversaries may gain root-level persistence despite factory resets. This means that even with security patches applied, breaches could go undetected, further increasing the potential impact.

On the same day, Ivanti released an update advisory clarifying that the above findings observed in CISA's lab, has not been seen in the wild or believed to be possible in a live customer environment. CISA and the other government agencies recommend that defenders run Ivanti's updated external ICT, released on 27 February.^[2]

Detection and Mitigation^{[1][2]}

IMDA recommends organisations to perform continual testing and validating of existing security controls to ensure detection and prevention against the MITRE ATT&CK techniques identified in this advisory:

- Search your networks for IOCs listed & block the IOCs provided if no business need to IP addresses.
- Assume that user and service account credentials stored within the affected Ivanti VPN appliances are likely compromised.
- Run the latest version of internal and external ICT after successful factory reset and patched to detect compromise. If compromise persists, discontinue the use of compromised appliances, and contact Ivanti for support.
- Limit outbound internet connections from SSL VPN appliances if no business need.
- Ensure SSL VPN appliances configured with Active Directory or LDAP authentication use low privilege accounts for the LDAP bind.
- Limit SSL VPN connections to unprivileged accounts only to limit the exposure of privileged account credentials.
- Strictly limit the use of Remote Desktop Protocols (RDP) and other remote access tools.
- Configure the Windows Registry to require User Account Control (UAC) approval for any PsExec operations.
- Refer to the MITRE ATT&CK techniques and validate security controls to create detection rules and deny processes related to these techniques if there is no business need.

IMDA encourages organisations to conduct thorough analysis to identify potential risks and assess their potential impact prior to deploying defensive measures.

Indicators of Compromise^[1]

Indicator	Type	Description
206.189.208[.]156	IP Address	DigitalOcean IP address tied to APT group UTA0178.
75.145.243[.]85	IP Address	UTA0178 IP address observed interacting with compromised device.
47.207.9[.]89	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
98.160.48[.]170	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
173.220.106[.]166	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
73.128.178[.]221	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
50.243.177[.]161	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
50.213.208[.]89	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
64.24.179[.]210	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
75.145.224[.]109	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
50.215.39[.]49	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
71.127.149[.]194	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
173.53.43[.]7	IP Address	UTA0178 IP address observed interacting with compromised device tied to Cyberoam proxy network.
146.0.228[.]66	IP Address	WARPWIRE variant C2 server
159.65.130[.]146	IP Address	WARPWIRE variant C2 server
8.137.112[.]245	IP Address	WARPWIRE variant C2 server
91.92.254[.]14	IP Address	WARPWIRE variant C2 server
186.179.39[.]235	IP Address	Mass exploitation activity
50.215.39[.]49	IP Address	Post-exploitation activity
45.61.136[.]14	IP Address	Post-exploitation activity
173.220.106[.]166	IP Address	Post-exploitation activity
88.119.169[.]227	IP-address	Malicious IP address
103.13.28[.]40	IP-address	Malicious IP address
46.8.68[.]100	IP-address	Malicious IP address
gpoaccess[.]com	Hostname	Suspected UTA0178 domain discovered via domain registration patterns.
webb-institute[.]com	Hostname	Suspected UTA0178 domain discovered via domain registration patterns.
symantke[.]com	Hostname	UTA0178 domain used to collect credentials from compromised devices.
symantke[.]com	Domain	WARPWIRE C2 server

miltonhouse[.]nl	Domain	WARPWIRE variant C2 server
entraide-internationale[.]fr	Domain	WARPWIRE variant C2 server
api.d-n-s[.]name	Domain	WARPWIRE variant C2 server
cpanel.netbar[.]org	Domain	WARPWIRE variant C2 server
clickcom[.]click	Domain	WARPWIRE variant C2 server
clicko[.]click	Domain	WARPWIRE variant C2 server
duorhytm[.]fun	Domain	WARPWIRE variant C2 server
line-api[.]com	Domain	WARPWIRE variant C2 server
arekaweb[.]com	Domain	WARPWIRE variant C2 server
ehangmun[.]com	Domain	WARPWIRE variant C2 server
secure-cama[.]com	Domain	WARPWIRE variant C2 server

SHA256/MD5 Hash	Description
ed4b855941d6d7e07aacf016a2402c4c870876a050a4a547af194f5a9b47945f	WIREFIRE web shell
3045f5b3d355a9ab26ab6f44cc831a83	CHAINLINE web shell
3d97f55a03ceb4f71671aa2ecf5b24e9	CHAINLINE web shell
2ec505088b942c234f39a37188e80d7a	LIGHTWIRE web shell
8eb042da6ba683ef1bae460af103cc44	WARPWIRE credential harvester variant
a739bd4c2b9f3679f43579711448786f	WARPWIRE credential harvester variant
a81813f70151a022ea1065b7f4d6b5ab	WARPWIRE credential harvester variant
d0c7a334a4d9dcd3c6335ae13bee59ea	WARPWIRE credential harvester variant
e8489983d73ed30a4240a14b1f161254	WARPWIRE credential harvester variant

YARA rules

```
rule apt_webshell_pl_complyshell: UTA0178
```

```
{
meta:
  author = "threatintel@volexity.com"
  date = "2023-12-13"
  description = "Detection for the COMPLYSHELL webshell."
  hash1 = "8bc8f4da98ee05c9d403d2cb76097818de0b524d90bea8ed846615e42cb031d2"
  os = "linux"
  os_arch = "all"
  report = "TIB-20231215"
  scan_context = "file,memory"
  last_modified = "2024-01-09T10:05Z"
  license = "See license at https://github.com/volexity/threat-intel/blob/main/LICENSE.txt"
  rule_id = 9995
  version = 4
```

```
strings:
```

```
$s = "eval{my $c=Crypt::RC4->new("
```

```
condition:
```

```
$s
```

```
}
```

```
rule apt_webshell_aspx_glasstoken: UTA0178
```

```
{
meta:
  author = "threatintel@volexity.com"
  date = "2023-12-12"
  description = "Detection for a custom webshell seen on external facing server. The webshell contains two functions, the first is to act as a Tunnel, using code borrowed from reGeorg, the second is custom code to execute arbitrary .NET code."
  hash1 = "26cbb54b1feb75fe008e36285334d747428f80aacdb57badf294e597f3e9430d"
  os = "win"
  os_arch = "all"
  report = "TIB-20231215"
  scan_context = "file,memory"
  last_modified = "2024-01-09T10:08Z"
  license = "See license at https://github.com/volexity/threat-intel/blob/main/LICENSE.txt"
  rule_id = 9994
  version = 5
```

```
strings:
```

```
$s1 = "=Convert.FromBase64String(System.Text.Encoding.Default.GetString(" ascii
$re = /Assembly\.Load\(errors\)\.CreateInstance\("[a-z0-9A-Z]{4,12}"\)\.GetHashCode\(\);/
```

```
condition:
```

```

for any i in (0..#s1):
(
$re in (@s1[i]..@s1[i]+512)
)
}

rule webshell_aspx_regeorg
{
meta:
author = "threatintel@volexity.com"
date = "2018-08-29"
description = "Detects the reGeorg webshell based on common strings in the webshell. May
also detect other webshells which borrow code from ReGeorg."
hash = "9d901f1a494ffa98d967ee6ee30a46402c12a807ce425d5f51252eb69941d988"
os = "win"
os_arch = "all"
reference = "https://github.com/L-codes/Neo-reGeorg/blob/master/templates/tunnel.aspx"
report = "TIB-20231215"
scan_context = "file,memory"
last_modified = "2024-01-09T10:04Z"
license = "See license at https://github.com/volexity/threat-intel/blob/main/LICENSE.txt"
rule_id = 410
version = 7

strings:
$a1 = "every office needs a tool like Georg" ascii
$a2 = "cmd = Request.QueryString.Get(\"cmd\")" ascii
$a3 = "exKak.Message" ascii

$proxy1 = "if (rkey != \"Content-Length\" && rkey != \"Transfer-Encoding\")"
$proxy_b1 = "StreamReader repBody = new StreamReader(response.GetResponseStream(),
Encoding.GetEncoding(\"UTF-8\"));" ascii
$proxy_b2 = "string rbody = repBody.ReadToEnd();" ascii
$proxy_b3 = "Response.AddHeader(\"Content-Length\", rbody.Length.ToString());" ascii

condition:
any of ($a*) or
$proxy1 or
all of ($proxy_b*)
}

rule hacktool_py_pysoxy
{
meta:
author = "threatintel@volexity.com"
date = "2024-01-09"
description = "SOCKS5 proxy tool used to relay connections."
hash1 = "e192932d834292478c9b1032543c53edfc2b252fdf7e27e4c438f4b249544eeb"

```

```
os = "all"
os_arch = "all"
reference = "https://github.com/MisterDaneel/pysoxy/blob/master/pysoxy.py"
report = "TIB-20240109"
scan_context = "file,memory"
last_modified = "2024-01-09T13:45Z"
license = "See license at https://github.com/volexity/threat-intel/blob/main/LICENSE.txt"
rule_id = 10065
version = 3
```

strings:

```
$s1 = "proxy_loop" ascii
$s2 = "connect_to_dst" ascii
$s3 = "request_client" ascii
$s4 = "subnegotiation_client" ascii
$s5 = "bind_port" ascii
```

condition:

```
all of them
}
```

rule apt_webshell_py_categorical: UTA0178

```
{
meta:
author = "threatintel@volexity.com"
date = "2024-01-18"
description = "Detection for the CATEGORICAL webshell."
os = "linux"
os_arch = "all"
scan_context = "file,memory"
severity = "critical"
```

strings:

```
$s1 = "exec(zlib.decompress(aes.decrypt(base64.b64decode" ascii
$s2 = "globals()[dskey].pop('result',None)" ascii
$s3 = "dsid=request.cookies.get('DSID'" ascii
```

condition:

```
any of ($s*)
}
```

MITRE ATT&CK Tactics and Techniques^[1]



Initial Access		
Technique Title	ID	Use
Exploit Public-Facing Applications	<u>T1190</u>	Threat actors use custom web shells planted on public facing applications which allows persistence in victims' environment.

Persistence

Technique Title	ID	Use
Valid Accounts	<u>T1078</u>	Threat actors leverage compromised accounts to laterally move within internal systems via RDP, SBD, and SSH.
Server Software Component: Web Shell	<u>T1505.003</u>	Threat actors may use web shells on internal- and external-facing web servers to establish persistent access to systems.

Execution		
Technique Title	ID	Use
Command and Scripting Interpreter: PowerShell	<u>T1059.001</u>	Threat actors leverage code execution from request parameters that are decoded from hex to base64 decoded, then passed to <code>Assembly.Load()</code> . Which is used to execute arbitrary powershell commands.
Exploitation for Client Execution	<u>T1203</u>	Threat actors exploit software vulnerabilities such as command-injection and achieve unauthenticated remote code execution (RCE).

References

1. [^ “Threat Actors Exploit Multiple Vulnerabilities in Ivanti Connect Secure and Policy Secure Gateways”](#) .
2. [^ “Enhanced External Integrity Checking Tool to Provide Additional Visibility and Protection for Customers Against Evolving Threat Actor Techniques in Relation to Previously Disclosed Vulnerabilities”](#) .