

# NETMANAGEIT

## Intelligence Report

# SQL Brute Force Leads to BlueSky Ransomware



# Table of contents

---

## Overview

---

● Description	4
● Confidence	4
● Content	5

---

## Entities

---

● Attack-Pattern	6
● Indicator	20
● Malware	27
● Vulnerability	28

---

## Observables

---

● StixFile	29
● Hostname	30
● IPv4-Addr	31

---

●	Url	32
---	-----	----

---

## External References

---

●	External References	33
---	---------------------	----

# Overview

## Description

Researchers observed an intrusion on a public-facing MSSQL Server, which resulted in BlueSky ransomware. First discovered in June 2022, BlueSky ransomware has code links to Conti and Babuk ransomware.

## Confidence

*This value represents the confidence in the correctness of the data contained within this report.*

15 / 100

# Content

N/A

# Attack-Pattern

**Name**

OS Credential Dumping

**ID**

T1003

**Description**

Adversaries may attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform [Lateral Movement](<https://attack.mitre.org/tactics/TA0008>) and access restricted information. Several of the tools mentioned in associated sub-techniques may be used by both adversaries and professional security testers. Additional custom tools likely exist as well.

**Name**

Valid Accounts

**ID**

T1078

**Description**

Adversaries may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Compromised

credentials may be used to bypass access controls placed on various resources on systems within the network and may even be used for persistent access to remote systems and externally available services, such as VPNs, Outlook Web Access, network devices, and remote desktop.(Citation: volexity\_0day\_sophos\_FW) Compromised credentials may also grant an adversary increased privilege to specific systems or access to restricted areas of the network. Adversaries may choose not to use malware or tools in conjunction with the legitimate access those credentials provide to make it harder to detect their presence. In some cases, adversaries may abuse inactive accounts: for example, those belonging to individuals who are no longer part of an organization. Using these accounts may allow the adversary to evade detection, as the original account user will not be present to identify any anomalous activity taking place on their account.(Citation: CISA MFA PrintNightmare) The overlap of permissions for local, domain, and cloud accounts across a network of systems is of concern because the adversary may be able to pivot across accounts and systems to reach a high level of access (i.e., domain or enterprise administrator) to bypass access controls set within the enterprise.(Citation: TechNet Credential Theft)

**Name**

Brute Force

**ID**

T1110

**Description**

Adversaries may use brute force techniques to gain access to accounts when passwords are unknown or when password hashes are obtained. Without knowledge of the password for an account or set of accounts, an adversary may systematically guess the password using a repetitive or iterative mechanism. Brute forcing passwords can take place via interaction with a service that will check the validity of those credentials or offline against previously acquired credential data, such as password hashes. Brute forcing credentials may take place at various points during a breach. For example, adversaries may attempt to brute force access to [Valid Accounts](<https://attack.mitre.org/techniques/T1078>) within a victim environment leveraging knowledge gathered from other post-compromise behaviors such as [OS Credential Dumping](<https://attack.mitre.org/techniques/T1003>), [Account Discovery](<https://attack.mitre.org/techniques/T1087>), or [Password Policy Discovery](<https://attack.mitre.org/techniques/T1201>). Adversaries may also combine brute forcing activity with behaviors such as [External Remote Services](<https://attack.mitre.org/techniques/T1133>) as part of Initial Access.

**Name**

Forced Authentication

**ID**

T1187

**Description**

Adversaries may gather credential material by invoking or forcing a user to automatically provide authentication information through a mechanism in which they can intercept. The Server Message Block (SMB) protocol is commonly used in Windows networks for authentication and communication between systems for access to resources and file sharing. When a Windows system attempts to connect to an SMB resource it will automatically attempt to authenticate and send credential information for the current user to the remote system. (Citation: Wikipedia Server Message Block) This behavior is typical in enterprise environments so that users do not need to enter credentials to access network resources. Web Distributed Authoring and Versioning (WebDAV) is also typically used by Windows systems as a backup protocol when SMB is blocked or fails. WebDAV is an extension of HTTP and will typically operate over TCP ports 80 and 443. (Citation: Didier Stevens WebDAV Traffic) (Citation: Microsoft Managing WebDAV Security) Adversaries may take advantage of this behavior to gain access to user account hashes through forced SMB/WebDAV authentication. An adversary can send an attachment to a user through spearphishing that contains a resource link to an external server controlled by the adversary (i.e. [Template Injection](<https://attack.mitre.org/techniques/T1221>)), or place a specially crafted file on navigation path for privileged accounts (e.g. .SCF file placed on desktop) or on a publicly accessible share to be accessed by victim(s). When the user's system accesses the untrusted resource it will attempt authentication and send information, including the user's hashed credentials, over SMB to the adversary controlled server. (Citation: GitHub Hashjacking) With access to the credential hash, an adversary can perform off-line [Brute Force](<https://attack.mitre.org/techniques/T1110>) cracking to gain access to plaintext credentials. (Citation: Cylance Redirect to SMB) There are several different ways this can occur. (Citation: Osanda Stealing NetNTLM Hashes) Some specifics from in-the-wild use include: \* A spearphishing attachment containing a document with a resource that is automatically loaded when the document is opened (i.e. [Template Injection](<https://attack.mitre.org/techniques/T1221>)). The document can include, for example, a request similar to `file[:]//[remote address]/Normal.dotm`` to trigger the SMB request. (Citation: US-CERT APT Energy Oct 2017) \* A modified .LNK or .SCF file with the icon filename pointing to an external reference such as `\\[remote address]\pic.png`` that will



force the system to load the resource when the icon is rendered to repeatedly gather credentials. (Citation: US-CERT APT Energy Oct 2017)

**Name**

Masquerading

**ID**

T1036

**Description**

Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusible system utilities to evade security monitoring is also a form of [Masquerading](<https://attack.mitre.org/techniques/T1036>). (Citation: LOLBAS Main Site) Masquerading may also include the use of [Proxy](<https://attack.mitre.org/techniques/T1090>) or VPNs to disguise IP addresses, which can allow adversaries to blend in with normal network traffic and bypass conditional access policies or anti-abuse protections.

**Name**

Process Injection

**ID**

T1055

**Description**

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources,

and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

**Name**

Scheduled Task/Job

**ID**

T1053

**Description**

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system.(Citation: TechNet Task Scheduler Security) Adversaries may use task scheduling to execute programs at system startup or on a scheduled basis for persistence. These mechanisms can also be abused to run a process under the context of a specified account (such as one with elevated permissions/privileges). Similar to [System Binary Proxy Execution](<https://attack.mitre.org/techniques/T1218>), adversaries have also abused task scheduling to potentially mask one-time execution under a trusted system process. (Citation: ProofPoint Serpent)

**Name**

Impair Defenses

**ID**

T1562

**Description**

Adversaries may maliciously modify components of a victim environment in order to hinder or disable defensive mechanisms. This not only involves impairing preventative defenses, such as firewalls and anti-virus, but also detection capabilities that defenders can use to audit activity and identify malicious behavior. This may also span both native defenses as well as supplemental capabilities installed by users and administrators. Adversaries may also impair routine operations that contribute to defensive hygiene, such as blocking users from logging out of a computer or stopping it from being shut down. These restrictions can further enable malicious operations as well as the continued propagation of incidents.(Citation: Emotet shutdown) Adversaries could also target event aggregation and analysis mechanisms, or otherwise disrupt these procedures by altering other system components.

**Name**

Modify Registry

**ID**

T1112

**Description**

Adversaries may interact with the Windows Registry to hide configuration information within Registry keys, remove information as part of cleaning up, or as part of other techniques to aid in persistence and execution. Access to specific areas of the Registry depends on account permissions, some requiring administrator-level access. The built-in Windows command-line utility [Reg](<https://attack.mitre.org/software/S0075>) may be used for local or remote Registry modification. (Citation: Microsoft Reg) Other tools may also be used, such as a remote access tool, which may contain functionality to interact with the Registry through the Windows API. Registry modifications may also include actions to hide keys, such as prepending key names with a null character, which will cause an error and/or be ignored when read via [Reg](<https://attack.mitre.org/software/S0075>) or other utilities using the Win32 API. (Citation: Microsoft Reghide NOV 2006) Adversaries may abuse these pseudo-hidden keys to conceal payloads/commands used to maintain persistence. (Citation: TrendMicro POWELIKS AUG 2014) (Citation: SpectorOps Hiding Reg Jul 2017) The

Registry of a remote system may be modified to aid in execution of files as part of lateral movement. It requires the remote Registry service to be running on the target system. (Citation: Microsoft Remote) Often [Valid Accounts](<https://attack.mitre.org/techniques/T1078>) are required, along with access to the remote system's [SMB/Windows Admin Shares](<https://attack.mitre.org/techniques/T1021/002>) for RPC communication.

**Name**

Data Encrypted for Impact

**ID**

T1486

**Description**

Adversaries may encrypt data on target systems or on large numbers of systems in a network to interrupt availability to system and network resources. They can attempt to render stored data inaccessible by encrypting files or data on local and remote drives and withholding access to a decryption key. This may be done in order to extract monetary compensation from a victim in exchange for decryption or a decryption key (ransomware) or to render data permanently inaccessible in cases where the key is not saved or transmitted.(Citation: US-CERT Ransomware 2016)(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017)(Citation: US-CERT SamSam 2018) In the case of ransomware, it is typical that common user files like Office documents, PDFs, images, videos, audio, text, and source code files will be encrypted (and often renamed and/or tagged with specific file markers). Adversaries may need to first employ other behaviors, such as [File and Directory Permissions Modification](<https://attack.mitre.org/techniques/T1222>) or [System Shutdown/Reboot](<https://attack.mitre.org/techniques/T1529>), in order to unlock and/or gain access to manipulate these files.(Citation: CarbonBlack Conti July 2020) In some cases, adversaries may encrypt critical system files, disk partitions, and the MBR.(Citation: US-CERT NotPetya 2017) To maximize impact on the target organization, malware designed for encrypting data may have worm-like features to propagate across a network by leveraging other attack techniques like [Valid Accounts](<https://attack.mitre.org/techniques/T1078>), [OS Credential Dumping](<https://attack.mitre.org/techniques/T1003>), and [SMB/Windows Admin Shares](<https://attack.mitre.org/techniques/T1021/002>).(Citation: FireEye WannaCry 2017)(Citation: US-CERT NotPetya 2017) Encryption malware may also leverage [Internal Defacement](<https://attack.mitre.org/techniques/T1491/001>), such as changing victim wallpapers, or otherwise intimidate victims by sending ransom notes or other messages to connected printers (known as "print bombing").(Citation: NHS Digital Egregor Nov 2020) In

cloud environments, storage objects within compromised accounts may also be encrypted.  
(Citation: Rhino S3 Ransomware Part 1)

**Name**

Create or Modify System Process

**ID**

T1543

**Description**

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services.(Citation: TechNet Services) On macOS, launchd processes known as [Launch Daemon](<https://attack.mitre.org/techniques/T1543/004>) and [Launch Agent](<https://attack.mitre.org/techniques/T1543/001>) are run to finish system initialization and load user specific parameters.(Citation: AppleDocs Launch Agent Daemons) Adversaries may install new services, daemons, or agents that can be configured to execute at startup or a repeatable interval in order to establish persistence. Similarly, adversaries may modify existing services, daemons, or agents to achieve the same effect. Services, daemons, or agents may be created with administrator privileges but executed under root/SYSTEM privileges. Adversaries may leverage this functionality to create or modify system processes in order to escalate privileges.(Citation: OSX Malware Detection)

**Name**

Obfuscated Files or Information

**ID**

T1027

**Description**

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) for [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](https://attack.mitre.org/techniques/T1027/010) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

**Name**

System Services

**ID**

T1569

**Description**

Adversaries may abuse system services or daemons to execute commands or programs. Adversaries can execute malicious content by interacting with or creating services either locally or remotely. Many services are set to run at boot, which can aid in achieving persistence ([Create or Modify System Process](https://attack.mitre.org/techniques/T1543)), but adversaries can also abuse services for one-time or temporary execution.

**Name**

## Command and Scripting Interpreter

**ID**

T1059

**Description**

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](<https://attack.mitre.org/techniques/T1059/004>) while Windows installations include the [Windows Command Shell](<https://attack.mitre.org/techniques/T1059/003>) and [PowerShell](<https://attack.mitre.org/techniques/T1059/001>). There are also cross-platform interpreters such as [Python](<https://attack.mitre.org/techniques/T1059/006>), as well as those commonly associated with client applications such as [JavaScript](<https://attack.mitre.org/techniques/T1059/007>) and [Visual Basic](<https://attack.mitre.org/techniques/T1059/005>). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](<https://attack.mitre.org/tactics/TA0001>) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](<https://attack.mitre.org/techniques/T1021>) in order to achieve remote Execution. (Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

**Name**

System Owner/User Discovery

**ID**

T1033

**Description**

Adversaries may attempt to identify the primary user, currently logged in user, set of users that commonly uses a system, or whether a user is actively using the system. They may do this, for example, by retrieving account usernames or by using [OS Credential Dumping] (<https://attack.mitre.org/techniques/T1003>). The information may be collected in a number of different ways using other Discovery techniques, because user and username details are prevalent throughout a system and include running process ownership, file/directory ownership, session information, and system logs. Adversaries may use the information from [System Owner/User Discovery](<https://attack.mitre.org/techniques/T1033>) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Various utilities and commands may acquire this information, including `whoami`. In macOS and Linux, the currently logged in user can be identified with `w` and `who`. On macOS the `dscl . list /Users | grep -v '_'` command can also be used to enumerate user accounts. Environment variables, such as `%USERNAME%` and `$USER`, may also be used to access this information. On network devices, [Network Device CLI](<https://attack.mitre.org/techniques/T1059/008>) commands such as `show users` and `show ssh` can be used to display users currently logged into the device. (Citation: `show_ssh_users_cmd_cisco`) (Citation: US-CERT TA18-106A Network Infrastructure Devices 2018)

**Name**

Remote Services

**ID**

T1021

**Description**

Adversaries may use [Valid Accounts](<https://attack.mitre.org/techniques/T1078>) to log into a service that accepts remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user. In an enterprise environment, servers and workstations can be organized into domains. Domains provide centralized identity management, allowing users to login using one set of credentials across the entire network. If an adversary is able to obtain a set of valid domain credentials, they could login to many different machines using remote access protocols such as secure shell (SSH) or remote desktop protocol (RDP). (Citation: SSH Secure Shell) (Citation: TechNet Remote Desktop Services) They could also login to accessible SaaS or IaaS services, such as those that federate their identities to the domain. Legitimate applications (such as [Software Deployment Tools](<https://attack.mitre.org/techniques/T1072>)) and other administrative



programs) may utilize [Remote Services](<https://attack.mitre.org/techniques/T1021>) to access remote hosts. For example, Apple Remote Desktop (ARD) on macOS is native software used for remote management. ARD leverages a blend of protocols, including [VNC](<https://attack.mitre.org/techniques/T1021/005>) to send the screen and control buffers and [SSH](<https://attack.mitre.org/techniques/T1021/004>) for secure file transfer. (Citation: Remote Management MDM macOS)(Citation: Kickstart Apple Remote Desktop commands)(Citation: Apple Remote Desktop Admin Guide 3.3) Adversaries can abuse applications such as ARD to gain remote code execution and perform lateral movement. In versions of macOS prior to 10.14, an adversary can escalate an SSH session to an ARD session which enables an adversary to accept TCC (Transparency, Consent, and Control) prompts without user interaction and gain access to data.(Citation: FireEye 2019 Apple Remote Desktop)(Citation: Lockboxx ARD 2019)(Citation: Kickstart Apple Remote Desktop commands)

**Name**

Application Layer Protocol

**ID**

T1071

**Description**

Adversaries may communicate using OSI application layer protocols to avoid detection/network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server. Adversaries may utilize many different protocols, including those used for web browsing, transferring files, electronic mail, or DNS. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are SMB, SSH, or RDP.

**Name**

System Information Discovery

**ID**

T1082

**Description**

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](https://attack.mitre.org/techniques/T1082) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Tools such as [Systeminfo](https://attack.mitre.org/software/S0096) can be used to gather detailed system information. If running with privileged access, a breakdown of system data can be gathered through the `systemsetup` configuration tool on macOS. As an example, adversaries with user-level access can execute the `df -aH` command to obtain currently mounted disks and associated freely available space. Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather detailed system information (e.g. `show version`).(Citation: US-CERT-TA18-106A) [System Information Discovery](https://attack.mitre.org/techniques/T1082) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment.(Citation: OSX.FairyTale)(Citation: 20 macOS Common Tools and Techniques) Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status of a particular instance or the model view of a virtual machine.(Citation: Amazon Describe Instance)(Citation: Google Instances Resource)(Citation: Microsoft Virtual Machine API)

**Name**

Network Share Discovery

**ID**

T1135

**Description**

Adversaries may look for folders and drives shared on remote systems as a means of identifying sources of information to gather as a precursor for Collection and to identify potential systems of interest for Lateral Movement. Networks often contain shared network drives and folders that enable users to access file directories on various systems

across a network. File sharing over a Windows network occurs over the SMB protocol. (Citation: Wikipedia Shared Resource) (Citation: TechNet Shared Folder) [Net](<https://attack.mitre.org/software/S0039>) can be used to query a remote system for available shared drives using the ``net view \\\reOTESystem`` command. It can also be used to query shared drives on the local system using ``net share``. For macOS, the ``sharing -l`` command lists all shared points used for smb services.

# Indicator

**Name**

3b463c94b52414cfaad61eccdac64ca84eaea1ab4be69f75834aaa7701ab5e7d0

**Description**

Empire\_Invoke\_PowerDump SHA256 of e7be97fb2200eb99805e39513304739a7a28b17e

**Pattern Type**

stix

**Pattern**

[file:hashes:'SHA-256' =  
'3b463c94b52414cfaad61eccdac64ca84eaea1ab4be69f75834aaa7701ab5e7d0']

**Name**

d4f4069b1c40a5b27ba0bc15c09dceb7035d054a022bb5d558850edfba0b9534

**Description**

Trojan:Win32/DorkBot.DU SHA256 of 07610f11d3b8ccb7b60cc8ad033dda6c7d3940c4

**Pattern Type**

stix

**Pattern**

```
[file:hashes:'SHA-256' =
'd4f4069b1c40a5b27ba0bc15c09dceb7035d054a022bb5d558850edfba0b9534']
```

**Name**

83.97.20.81

**Description**

```
**ISP:** M247 Europe SRL **OS:** None ----- Hostnames: -
asd.s7610rir.pw - 81.20.97.83.ro.ovo.sc ----- Domains: - ovo.sc - s7610rir.pw
----- Services: **22:** ~~~ SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubuntu0.4 Key
type: ecdsa-sha2-nistp256 Key:
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKpdDMYUa/1gdt7/
pBsr95DL pkx81TvWFR8r4LyXFR8GRpyXCpF0+Cvk/pkQggiknY7jvQlvTD4HnNkQuFLDV+U=
Fingerprint: d7:84:b1:87:f9:4f:b3:02:8f:c0:db:9c:0e:cb:b5:9e Kex Algorithms: curve25519-sha256
curve25519-sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521
sntrup761x25519-sha512@openssh.com diffie-hellman-group-exchange-sha256 diffie-
hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256
Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ecdsa-sha2-nistp256 ssh-ed25519
Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr
aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-
etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com
hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com
umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression
Algorithms: none zlib@openssh.com ~~~ ----- **80:** ~~~ HTTP/1.1 404 Not Found
Server: nginx Date: Tue, 05 Dec 2023 15:31:19 GMT Content-Type: text/html;
charset=iso-8859-1 Content-Length: 196 Connection: keep-alive ~~~ ----- **443:**
~~~ HTTP/1.1 404 Not Found Server: nginx Date: Thu, 30 Nov 2023 20:50:55 GMT Content-Type:
text/html; charset=iso-8859-1 Content-Length: 196 Connection: keep-alive ~~~ HEARTBLEED:
2023/11/30 20:51:20 83.97.20.81:443 - ERROR: dial tcp 83.97.20.81:443: connection refused
-----
```

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '83.97.20.81']

**Name**

74b6d14e35ff51fe47e169e76b4732b9f157cd7e537a2ca587c58dbdb15c624f

**Pattern Type**

stix

**Pattern**

[file:hashes!'SHA-256' =  
'74b6d14e35ff51fe47e169e76b4732b9f157cd7e537a2ca587c58dbdb15c624f']

**Name**

https://asd.s7610rir.pw/win/checking.hta

**Pattern Type**

stix

**Pattern**

[url:value = 'https://asd.s7610rir.pw/win/checking.hta']

**Name**

http://asq.r77vh0.pw/win/checking.hta

**Pattern Type**

stix

**Pattern**

[url:value = 'http://asq.r77vh0.pw/win/checking.hta']

**Name**

f955eeb3a464685eaac96744964134e49e849a03fc910454faaff2109c378b0b

**Pattern Type**

stix

**Pattern**

[file:hashes!'SHA-256' =  
'f955eeb3a464685eaac96744964134e49e849a03fc910454faaff2109c378b0b']

**Name**

5.188.86.237

**Description**

AsyncRAT botnet C2 server (confidence level: 100%)

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '5.188.86.237']

**Name**

asq.d6shiiwz.pw

**Pattern Type**

stix

**Pattern**

[hostname:value = 'asq.d6shiiwz.pw']

**Name**

asq.r77vh0.pw

**Pattern Type**

stix

**Pattern**

[hostname:value = 'asq.r77vh0.pw']

**Name**

https://asq.d6shiiwz.pw/win/hssl/d6.hta

**Pattern Type**

stix

**Pattern**



[url:value = 'https://asq.d6shiiwz.pw/win/hssl/d6.hta']

**Name**

https://asq.r77vh0.pw/win/hssl/r7.hta

**Pattern Type**

stix

**Pattern**

[url:value = 'https://asq.r77vh0.pw/win/hssl/r7.hta']

**Name**

asd.s7610rir.pw

**Pattern Type**

stix

**Pattern**

[hostname:value = 'asd.s7610rir.pw']

**Name**

11bd2c9f9e2397c9a16e0990e4ed2cf0679498fe0fd418a3dfdac60b5c160ee5

**Description**

SHA256 of d25340ae8e92a6d29f599fef426a2bc1b5217299

**Pattern Type**

stix

**Pattern**

[file:hashes!'SHA-256' =  
'11bd2c9f9e2397c9a16e0990e4ed2cf0679498fe0fd418a3dfdac60b5c160ee5']

**Name**

35b95496b243541d5ad3667f4aabe2ed00066ba8b69b82f10dd1186872ce4be2

**Description**

TEL:Trojan:PowerShell/Wemaeye.B SHA256 of e938646862477e598fcda20d0b7551863f8b651c

**Pattern Type**

stix

**Pattern**

[file:hashes!'SHA-256' =  
'35b95496b243541d5ad3667f4aabe2ed00066ba8b69b82f10dd1186872ce4be2']

# Malware

**Name**

Daphne

**Name**

BlueSky

**Name**

Tor2Mine

**Name**

Cobalt Strike

**Description**

[Cobalt Strike](<https://attack.mitre.org/software/S0154>) is a commercial, full-featured, remote access tool that bills itself as “adversary simulation software designed to execute targeted attacks and emulate the post-exploitation actions of advanced threat actors”. Cobalt Strike’s interactive post-exploit capabilities cover the full range of ATT&CK tactics, all executed within a single, integrated system.(Citation: cobaltstrike manual) In addition to its own capabilities, [Cobalt Strike](<https://attack.mitre.org/software/S0154>) leverages the capabilities of other well-known tools such as Metasploit and [Mimikatz](<https://attack.mitre.org/software/S0002>). (Citation: cobaltstrike manual)

# Vulnerability

**Name**

CVE-2023-27350

**Description**

PaperCut MF/NG contains an improper access control vulnerability within the SetupCompleted class that allows authentication bypass and code execution in the context of system.

# StixFile

**Value**

11bd2c9f9e2397c9a16e0990e4ed2cf0679498fe0fd418a3dfdac60b5c160ee5

35b95496b243541d5ad3667f4aabe2ed00066ba8b69b82f10dd1186872ce4be2

74b6d14e35ff51fe47e169e76b4732b9f157cd7e537a2ca587c58dbdb15c624f

3b463c94b52414cfaad61ecdac64ca84eaea1ab4be69f75834aaa7701ab5e7d0

f955eeb3a464685eaac96744964134e49e849a03fc910454faaff2109c378b0b

d4f4069b1c40a5b27ba0bc15c09dceb7035d054a022bb5d558850edfba0b9534

# Hostname

**Value**

asq.r77vh0.pw

asd.s7610rir.pw

asq.d6shiiwz.pw

# IPv4-Addr

## Value

83.97.20.81

5.188.86.237

# Url

**Value**

<https://asd.s7610rir.pw/win/checking.hta>

<https://asq.r77vh0.pw/win/hssl/r7.hta>

<http://asq.r77vh0.pw/win/checking.hta>

<https://asq.d6shiiwz.pw/win/hssl/d6.hta>



# External References

- 
- <https://otx.alienvault.com/pulse/65707ab6e66cbcb43bd4f250>
- 
- <https://thefirreport.com/2023/12/04/sql-brute-force-leads-to-bluesky-ransomware/>