

Echo T&E

Penetration Test

Sample Report

Sample Company
Attn. Sample Customer
Broadway
New York City

Date: April 21, 2024
Report Version: 1.0

Echo T&E
Florida, USA

Table of Contents

1 Document Control	4
1.1 Team	4
1.2 List of Changes	4
2 Executive Summary	5
2.1 Overview	5
2.2 Identified Vulnerabilities	5
3 Methodology	7
3.1 Tooling	7
3.2 Objective	8
3.3 Scope	8
3.4 User Accounts and Permissions	10
4 Findings	11
C1: Authentication Bypass / Remote Code Execution: CVE-2024-27198 / CVE-2024-27199	11
H1: Weak and Recycled Credentials	14
H2: Information Disclosure (Admin+)	16
H3: Local Privilege Escalation via Portainer Service Misconfiguration	18
I1: Verbose Error Messages	22
5 Disclaimer and Legal	23
A Appendix	24
A.1 CVSS Scoring Diagram	24
A.2 Kill Chain	24



1 Document Control

1.1 Team

Contact	Details	Role
Jeremy Laratro	E-Mail: holly.kauri6625@eagereverest.com	Test Engineer

1.2 List of Changes

Version	Description	Date
1.0	Started report	Apr 21, 2024
1.1	Finalized body	Apr 22, 2024
1.2	Finalized vulnerabilities and findings	Apr 23, 2024

2 Executive Summary

2.1 Overview

The primary objective of this testing engagement was to assess the customer's security posture. This includes real-world cyber attack simulations and general security testing of the in-scope assets and digital infrastructure. Various target types are tested to assess any [in-scope] vectors of attack that may impact confidentiality, integrity, and accessibility of the systems being tested. The desired outcome of the testing engagement is to understand the current cybersecurity posture of the systems under test and subsequently, if necessary, how those systems can be better secured against a vast array of modern attack vectors faced in today's digital world.

2.2 Identified Vulnerabilities

#	CVSS	Description	Page
C1	9.3	Authentication Bypass / Remote Code Execution: CVE-2024-27198 / CVE-2024-27199	11
H 1	8.8	Weak and Recycled Credentials	14
H 2	8.7	Information Disclosure (Admin+)	16
H 3	7.2	Local Privilege Escalation via Portainer Service Misconfiguration	18
I1	0.0	Verbose Error Messages	22

Vulnerability Overview

In the course of this penetration test **1 Critical**, **3 High** and **1 Info** vulnerabilities were identified:

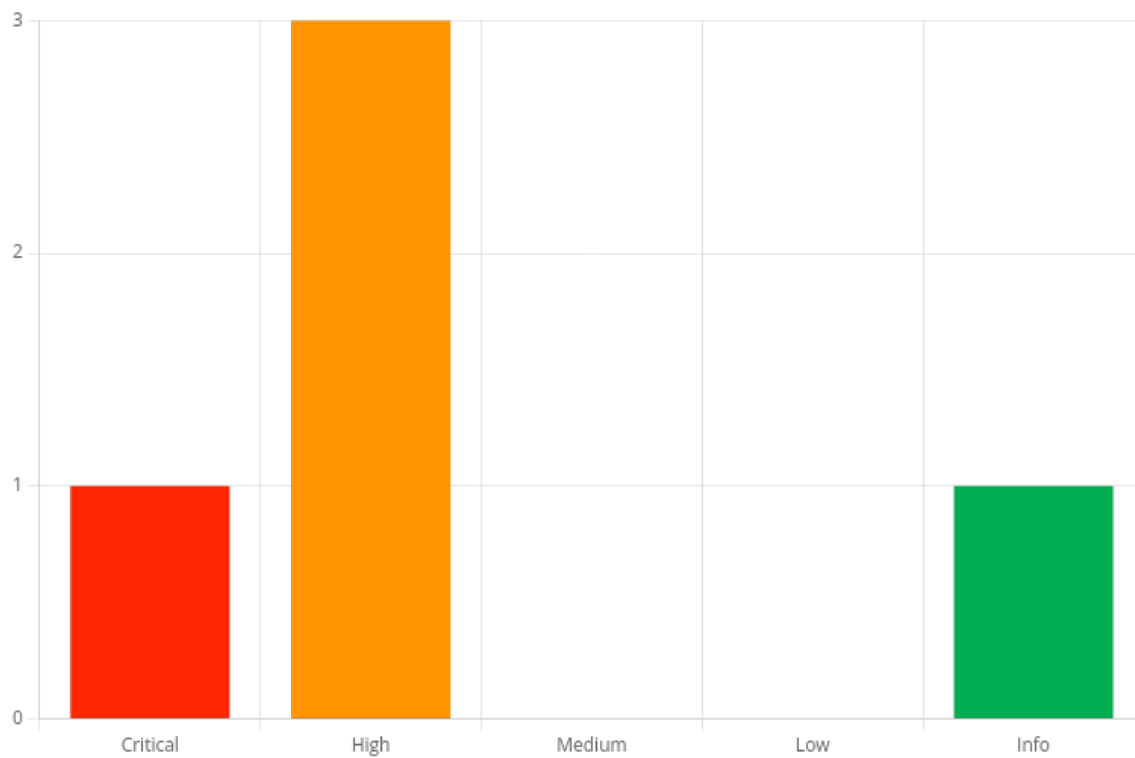


Figure 1 - Distribution of identified vulnerabilities

3 Methodology

All testing is performed with general objectives, depending on business needs, using industry-standard and accepted frameworks. These include OWASP Top 10, MITRE ATT@CK, and the CIA triad. Testing is primarily performed manually and is supplemented with automated technologies, including but not limited to, industry-standard tools as well as custom-written scripts in order to best understand and map the attack surface of the systems being tested.

The testing process involves the following general phases: planning and reconnaissance, active assessment, exploitation, post-exploitation, and reporting. The process is highly fluid and the engineer(s) performing the test will move between these phases as necessary for the duration of the penetration testing engagement.

Testing is performed throughout the engagement period, regardless of findings. In the case of critical findings that could potentially open up additional pathways not within the original scope, the engineer(s) will reach out to the customer to discuss whether an expansion of the scope is wanted, possible, and/or useful prior to moving on.

3.1 Tooling

This section provides a brief list of key penetration testing tools that were utilized or considered in various phases of the security assessment. The list below is not an exhaustive list and testing is not constrained to the tools listed below.

1. Reconnaissance and Information Gathering:

- **Nmap:** Network mapping and security auditing.
- **Shodan:** Internet-connected device search engine.
- **RustScan:** Modern port scanner that allows for faster and more efficient network enumeration.
- **Wireshark:** Network protocol analyzer for network troubleshooting and analysis.
- **Bloodhound:** Uses graph theory to reveal the hidden and often unintended relationships within an Active Directory environment.

2. Vulnerability Scanning and Web Testing:

- **Burp Suite Pro:** Advanced set of tools for security testing of web applications.
- **Gobuster:** Directory, file, and DNS enumeration tool using brute-forcing.
- **WFuzz:** Web application security fuzzer.
- **FFuf (Fuzz Faster U Fool):** High-speed web fuzzer.
- **WPScan:** WordPress vulnerability scanner.
- **SQLmap:** Automated tool for SQL injection and database takeover.

3. Exploitation Tools:

- **Metasploit:** Comprehensive exploitation toolset.
- **Impacket:** Collection of Python classes for working with network protocols.
- **Mythic:** Command and control framework that emphasizes a diverse set of communication strategies.
- **Evil-WinRM:** The ultimate WinRM shell for hacking/pentesting.

- **Netexec (CrackMapExec):** Swiss army knife for pentesting networks.
- **Responder:** LLMNR, NBT-NS and MDNS poisoner.

4. Password Cracking and Reverse Engineering:

- **Mimikatz:** Tool to extract plaintexts passwords, hashes, PIN codes, and Kerberos tickets from memory.
- **Hashcat:** Advanced password recovery utility.
- **Ghidra:** Software reverse engineering (SRE) framework developed by NSA's Research Directorate.
- **GDB (GNU Debugger):** Portable debugger that runs on many Unix-like systems.

Each tool is selected based on its performance, reliability, and capability to integrate seamlessly with other components of a comprehensive security assessment. These tools are pivotal for effectively identifying vulnerabilities, simulating real-world attacks, analyzing network traffic, and enhancing the security posture of an organization.

3.2 Objective

Utilizing the methodology explained, the penetration testing engagement also involves main objectives defined by customer needs. The main objectives of the penetration test are as follows:

- Map the attack surface of customer systems and infrastructure.
- Identify any weaknesses against modern, real-world threats.
- Assess resilience against simulated cyber attacks (excluding DoS).
- Find positive aspects of the systems and infrastructure in relation to cyber security.
- Compile this information in order to best inform the customer on required changes, fixes, urgency, as well as short and long-term strategies when applicable.

3.3 Scope

The execution portion of the penetration testing engagement was performed from **Apr 19, 2024 to Apr 21, 2024**. The engagement was carried out over a period of **5 days**.

The scope of this penetration testing engagement is defined to include specific targets within the client's network. This focused approach ensures that testing is both efficient and within the agreed contractual and legal boundaries. The targets specified for this penetration test are:

Network Scope:

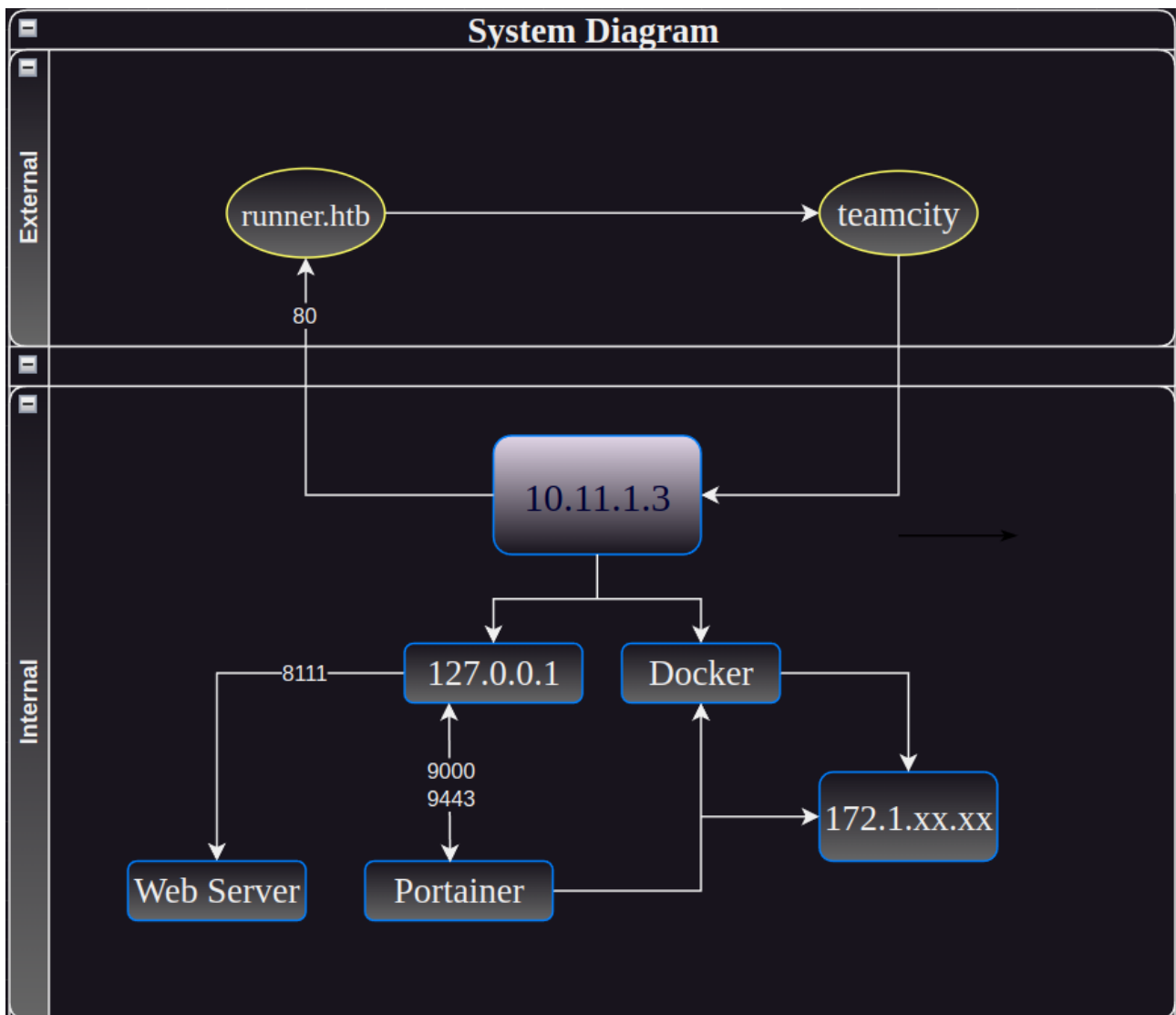
- **IP Address:**
 - 10.11.1.3

Domain Scope:

- **Domain:**
 - *.runner.htb - All subdomains under runner.htb are included in the scope.

System Overview

The system being tested is comprised of a single linux server which serves multiple purposes; hosting a main business domain, an employee subdomain, and an internal network comprised of various containers and internal web servers and services.



Test Objectives:

The testing will focus on identifying security vulnerabilities in the included IP address and domain, with particular attention to:

- System and network vulnerabilities.
- Web application security issues.
- Data leakage.
- Authentication and authorization flaws.
- Any misconfigurations that could be exploited by malicious actors.

Constraints:

- All testing is to be conducted in a manner that avoids any disruption to normal business operations.
- Testing will be limited to the assets defined within the scope. No other systems, IP addresses, or domains will be tested without prior written approval.

- The test engineer(s) will adhere to the highest ethical standards, using only non-destructive methods to ensure the integrity and availability of the client's systems and data.

Compliance:

The penetration test will comply with all applicable laws and industry standards, ensuring that all activities are conducted legally and ethically.

This scoped approach ensures a thorough assessment of the specified targets while maintaining the integrity and confidentiality of the client's broader network infrastructure.

3.4 User Accounts and Permissions

Provided Users

- None; testing was performed in an entirely black-box manner.

4 Findings

C1: Authentication Bypass / Remote Code Execution: CVE-2024-27198 / CVE-2024-27199	
Score	9.3 (Critical)
Vector string	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N/E:F/RL:O/RC:C/CR:H/IR:H/AR:L/MAV:N/MAC:L/MPR:N/MUI:N
Target	teamcity.runner.htb
References	<ul style="list-style-type: none"> • https%3A%2F%2Fwww.rapid7.com%2Fblog%2Fpost%2F2024%2F03%2F04%2Fetr-cve-2024-27198-and-cve-2024-27199-jetbrains-teamcity-multiple-authentication-bypass-vulnerabilities-fixed%2F • https%3A%2F%2Fblog.jetbrains.com%2Fteamcity%2F2024%2F02%2Fcritical-security-issue-affecting-teamcity-on-premises-cve-2024-23917%2F • https%3A%2F%2Fwww.cisa.gov%2Fnews-events%2Falerts%2F2024%2F03%2F07%2Fcisa-adds-one-known-exploited-jetbrains-vulnerability-cve-2024-27198-catalog

Overview

The server was confirmed to be vulnerable to two related CVEs, shown below:

- CVE-2024-27198 is an authentication bypass vulnerability in the web component of TeamCity that arises from an alternative path issue () and has a CVSS base score of 9.8 (Critical).
- CVE-2024-27199 is an authentication bypass vulnerability in the web component of TeamCity that arises from a path traversal issue () and has a CVSS base score of 7.3 (High).

Details

The TeamCity portal was exposed publicly at teamcity.runner.htb and was confirmed to be vulnerable/not patched. The CVE was successfully executed and later chained with other vulnerabilities to demonstrate impact. This is an urgent, critical finding that requires immediate action. Multiple public proof-of-concept scripts are available on the internet to exploit this vulnerability.

The TeamCity portal was initially found by fuzzing for subdomains of the root domain using wfuzz. The TeamCity portal shown below is considered an on-premises service, meaning that the service is not recommended to be exposed publicly and should rather be solely accessible via an on-premises network or internal network, for example, accessible only via VPN.

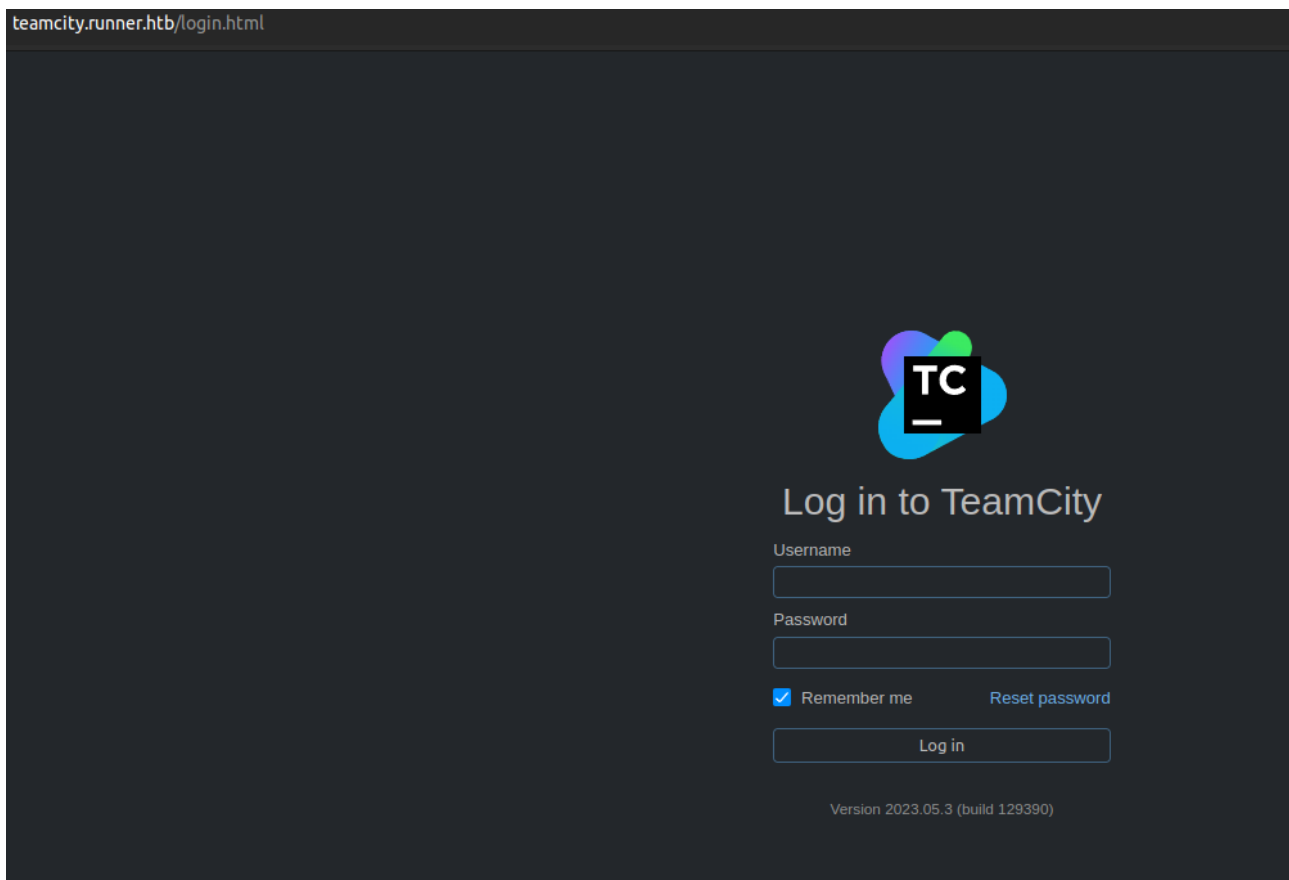


Figure 1: Exposed landing page discovered during fuzzing of the webserver.

After finding the exposed portal, the version number shown at the bottom of the page was used in subsequent research, which resulted in a positive identification of the service's vulnerable status.

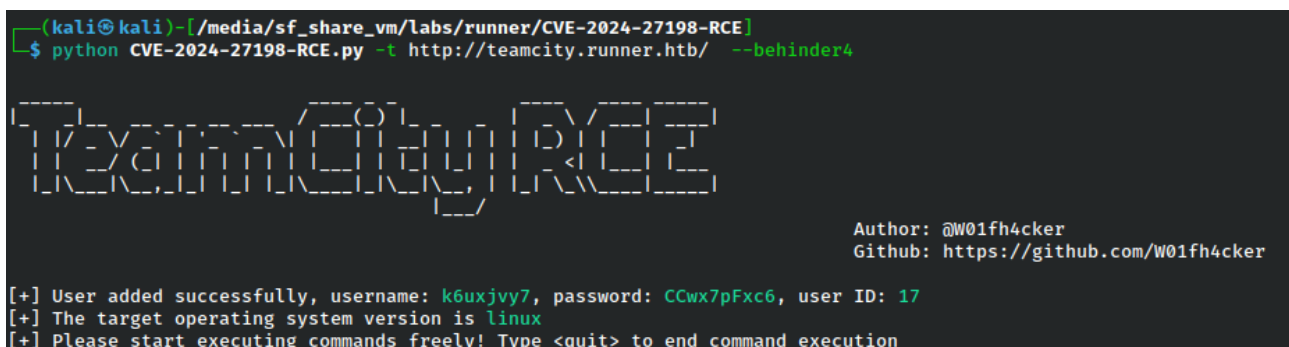


Figure 2: Proof of concept successful execution.

The proof-of-concept (PoC) script works by adding an administrative user and also allows for execution of arbitrary code on the server. After execution, the creation of new admin accounts was confirmed by logging in using the newly created credentials, leading to the admin portal and admin rights.





+ Create user account			4 users		
<input type="checkbox"/>	Username ^		Name ⇅		Email ⇅
<input type="checkbox"/>	admin		 John		john@runner.htb
<input type="checkbox"/>	h454nsec6608		 N/A		N/A
<input type="checkbox"/>	matthew		 Matthew		matthew@runner.htb
<input type="checkbox"/>	q480tbnc		 N/A		q480tbnc@example.com

Figure 3: Subsequent access to admin portal.

Recommendation

Short-Term: Remediation of the discovered vulnerability can be achieved by patching the server immediately.

Long-Term: The service should be repositioned so that it is not accessible via commercial internet. Segmentation of critical resources such as this is essential to the security of the network. If remote access is necessary, the service should be behind an access control mechanism. Options include segmenting the service so that it is accessible only via a secure VPN connection to prevent public access or a private cloud server with IAM for microsegmentation.

H1: Weak and Recycled Credentials	
Score	8.8 (High)
Vector string	CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N/E:P/RL:O/RC:C/CR:H/IR:H/AR:H/MAV:N/MAC:L/MPR:N/MUI:N/MS:U/MC:H/MI:H/MA:N
Target	teamcity.runner.htb
References	<ul style="list-style-type: none"> • https%3A%2F%2Fowasp.org%2Fwww-project-web-security-testing-guide%2Flatest%2F4-Web_Application_Security_Testing%2F04-Authentication_Testing%2F07-Testing_for_Weak_Password_Policy • https%3A%2F%2Fwww.acunetix.com%2Fvulnerabilities%2Fweb%2Fweak-password%2F

Overview

A weak password is short, common, a system default, or something that could be rapidly guessed by executing a brute force attack using a subset of all possible passwords, such as words in the dictionary, proper names, words based on the user name or common variations on these themes. In this case, a user was found to be using an extremely simple password and reusing it across multiple scopes (web and internal network).

Details

A user was found to be using a weak and reused password both externally and internally. Discovery of the weak credentials was via logs in the Administrator portal, however, due to the small size and simple schema of the password, bruteforcing was also possible using common, publicly accessible password dictionaries. The discovered password was easily brute-forced using Hashcat, a bruteforcing tool, and despite hashing with bcrypt, a relatively strong hashing algorithm for password storage in databases, the password was "cracked" in less than five minutes.

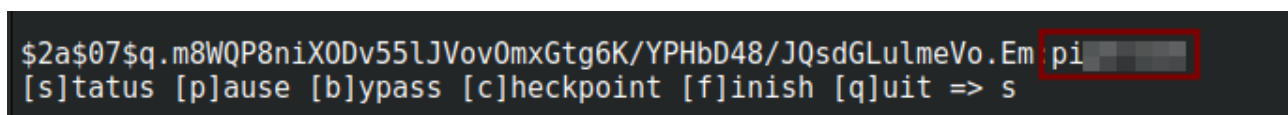


Figure 4: Password successfully cracked using Hashcat.

The password was valid for both the TeamCity portal and well as an internal Portainer portal. The external user on the TeamCity portal was a normal, non-admin user. While the impact of compromise of the external account was not extremely high, the ability to crack this password and the reuse of the password played a pivotal role in the kill chain that led to complete system compromise.

Recommendation

Short-Term: Remediation can be achieved by implementing a robust password policy that requires users to have strong passwords of reasonable length and character complexity, for example, a



minimum of 10 characters with two numbers and two special characters. Additionally, the password policy should implement maximum password age.

Long-Term: Implementation of MFA infrastructure is highly recommended. MFA can be achieved via various avenues, including code-based MFA like Google Authenticator and DUO, SMS-based MFA, hardware-based MFA like YubiKey, as well as the utilization of PKI infrastructure with physical cards that contain security certificates. Other options include passwordless authentication services, such as Beyond Identity. All options, if implemented correctly, would provide a significant enhancement of the company's data.

H2: Information Disclosure (Admin+)	
Score	8.7 (High)
Vector string	CVSS:4.0/AV:N/AC:L/AT:P/PR:H/UI:N/VC:H/VI:H/VA:N/SC:H/SI:H/SA:N/CR:H/IR:H/MAC:L/MPR:H/MUI:N/AU:N/R:A/V:C/RE:L/U:Amber
Target	teamcity.runner.htb
References	https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure

Overview

Summary

An information disclosure is the accidental, unintentional, or improper leak of information that is considered privileged or confidential. In this case, the disclosure is severely limited as it requires administrator access. However, despite the high privileges needed, such disclosures of sensitive information should be avoided as in the case of this test, the compromise of an admin account led to complete system takeover due to this specific issue.

Details

Description

An information disclosure requiring administrative privileges was discovered in the administrator portal of TeamCity. The information disclosed included an ssh private key, database logs that contained hashed user passwords, site backups, and other relevant server information. The disclosure of both the password hashes and the SSH private key were later leveraged in a multi-step process that concluded in complete system takeover.

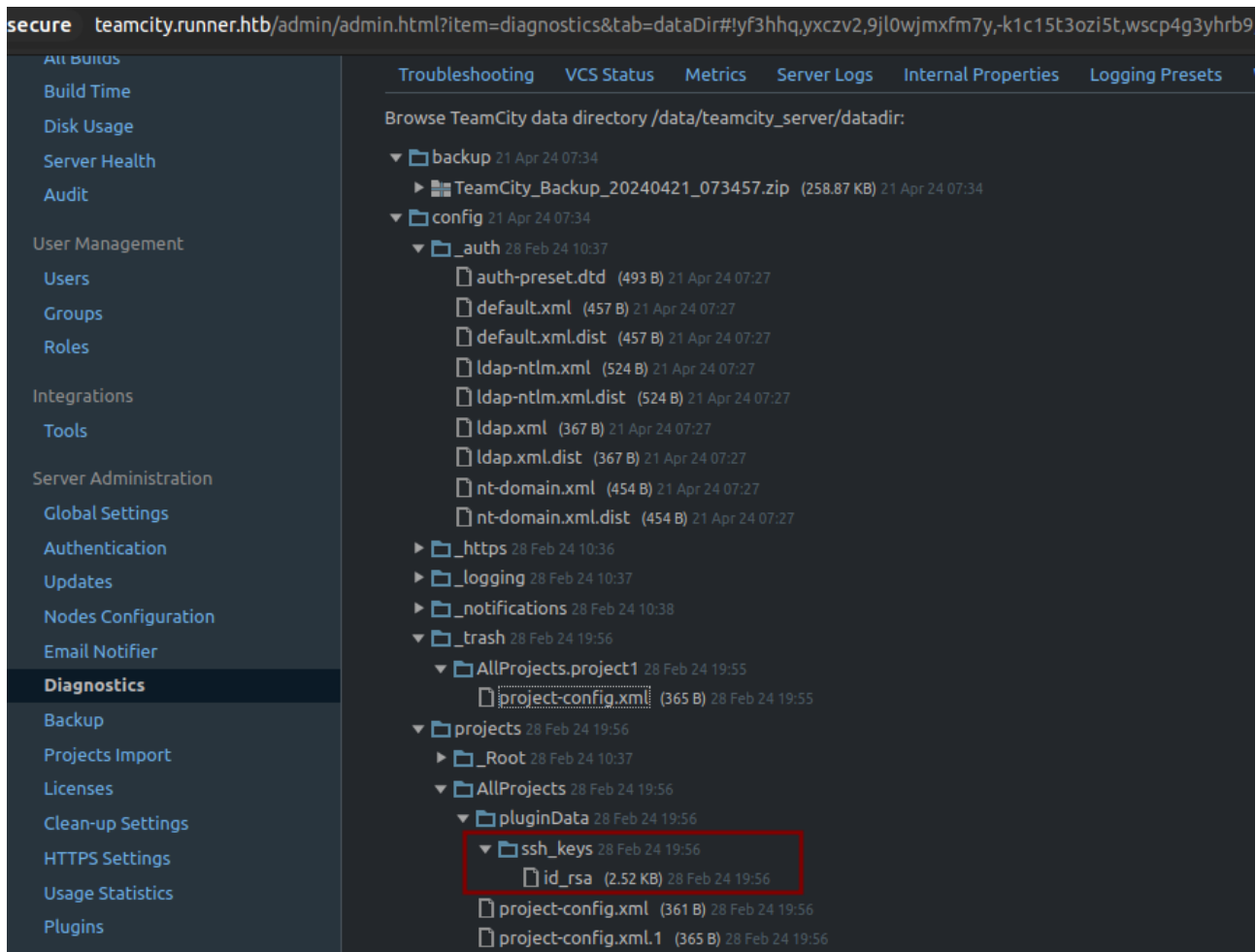


Figure 5: Disclosure of current SSH private keys via accessible, unencrypted backup.

Recommendation

Short-Term: The SSH keys should be immediately changed and removed. The other information within the backups should be assessed and removed if not necessary to reduce the chances of leaking other information should an administrator account be compromised. Additionally, future backups should be both encrypted and configured to exclude any directories containing ssh keys.

Long-Term: n/a

H3: Local Privilege Escalation via Portainer Service Misconfiguration

Score	7.2 (High)
Vector string	CVSS:3.1/AV:L/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H/E:P/RC:C/CR:H/IR:H/AR:H/MAV:L/MAC:H/MPR:H/MUI:N/MS:C/MC:H/MI:H/MA:H
Target	10.11.1.3
References	https://rioasmara.com/2021/08/15/use-portainer-for-privilege-escalation/

Overview

The Portainer service allowed the Portainer admin (compromised account), who is not a privileged system user, to mount the host filesystem and execute commands as the root user.

- Weak Credentials: An authentication form relies on weak credentials and can be easily bruteforced or cracked by an attacker
- Service Misconfiguration: An internal misconfigured service can lead to escalation of privileges
- Privilege Escalation: A vulnerability in which an attacker can escalate their privilege to either another user on the same level (horizontal privilege escalation) or to an user with higher privileges

Details

After gaining access to the system via the ssh key obtained previously, a cursory search of the system resulted in the discovery that the Portainer service was running locally.

```
john@runner:~$ ls /opt
containerd portainer
john@runner:~$ ifconfig
br-21746deff6ac: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.███ netmask 255.255.0.0 broadcast 172.███
    ether 02:4███ txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.███ netmask 255.255.0.0 broadcast 172.███
    inet6 fe80::███ prefixlen 64 scopeid 0x20<link>
    ether 02:███ txqueuelen 0 (Ethernet)
    RX packets 129032 bytes 89725069 (89.7 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 151839 bytes 26526473 (26.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 6: Discovery of the Portainer service via server enumeration.

A simple follow-up command was executed to find the internal port where the Portainer service was accessible at.

```
john@runner:~$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address

tcp        0      0 0.0.0.0:80
tcp        0      0 0.0.0.0:22
tcp        0      0 127.0.0.53:53
tcp        0      0 127.0.0.1:5005
tcp        0      0 127.0.0.1:9000
tcp        0      0 127.0.0.1:9443
tcp        0      0 127.0.0.1:8111
```

Figure 7: Usage of netstat, a native Linux tool, to discover the web portal port of the Portainer service.

Using SSH dynamic forwarding, the engineer was able to access the internal Portainer portal and authentication was attempted using the previously compromised user's credentials, which turned out to be successful.

Users		
Name ↓ ↑	Role ↓ ↑	Authentication ↓ ↑
r	team leader	Internal

Figure 8: Compromised user enumeration within the Portainer webservice.

The compromised user was also found to have elevated privileges within this service. The compromised user's Portainer privileges were then leveraged by creating a new container with an interactive shell, root user, and the target volume mapped within /mnt.

Volume mapping
+ map additional volume

container	/mnt/root
→ volume	kiberjen - local

Figure 9: Volume mapping step of privilege escalation.

Working Dir
/
User
root

Console
☒ Interactive & TTY (-i -t)
☐ TTY (-t)
☐ Interactive (-i)
☐ None

Figure 10: TTY console enabled for privilege escalation.

Once the above settings were established, the container was deployed and the console was accessed. As expected, the console was running as the root user and the target filesystem with all of the company data and information was easily accessible by navigating to the /mnt directory.

```

root@b54f6a5509d1:/# cd /mnt/root
root@b54f6a5509d1:/mnt/root# ls
bin boot data dev etc home lib lib32 lib64 libx32
root@b54f6a5509d1:/mnt/root# cd /root; whoami; id
root
uid=0(root) gid=0(root) groups=0(root)
root@b54f6a5509d1:~#

```

Figure 11: Successful escalation from normal user to root user.

It was also possible to mount other interfaces; if the system was the main hypervisor for a virtualized server farm, the consequence would have been compromise of all servers. The result of this misconfiguration in case was full system compromise of both the virtualized server and the host.

Recommendation

Short-Term: Reconfigure account permissions so that the Portainer service cannot mount the host file system.

Long-Term: A robust account segmentation and password policy (as described in previous finding) should be implemented. A non-root system user should not be able to easily bypass account restrictions and escalate to root user via the Portainer service. Only users who should have access to the system root account should be able to mount the host and other critical filesystems via the Portainer service and subsequently execute shell commands as root. Additionally, accounts of privileged status should not be integrated with accounts of low permission elsewhere. Segmentation by risk category is recommended to resolve these issues.

I1: Verbose Error Messages	
Score	0.0 (Info)
Vector string	N/A
Target	10.11.1.3
References	https://owasp.org/www-project-proactive-controls/v3/en/c10-errors-exceptions

Overview

The clone feature within teamcity, under the project section, returns a verbose error that contains potentially useful information of the underlying system. Improper handling of errors can introduce a variety of security problems for a web site. The most common problem is when detailed internal error messages such as stack traces, database dumps, and error codes are displayed to the user (hacker). These messages reveal implementation details that should never be revealed. Such details can provide hackers important clues on potential flaws in the site and such messages are also disturbing to normal users.

Details

The application returned the following error message when entering any arbitrary or invalid git path:

```
Failed to perform checkout on agent: '/usr/bin/git -c core.askpass=/opt/app/runner/
teamcity/pass2043620868177620056 fetch --progress origin +refs/heads/master:refs/heads/
master' command failed.
exit code: 128
```

The error message provides the path of the teamcity directory within opt, and provides potentially useful information about the underlying server's file system and structure.

Recommendation

Short-Term: Verbose error messages such as the one shown should be limited to debugging within development environments and never enabled or allowed within production environments. It is recommended that the error is edited to something vague that does not contain any clues about the server's filesystem.

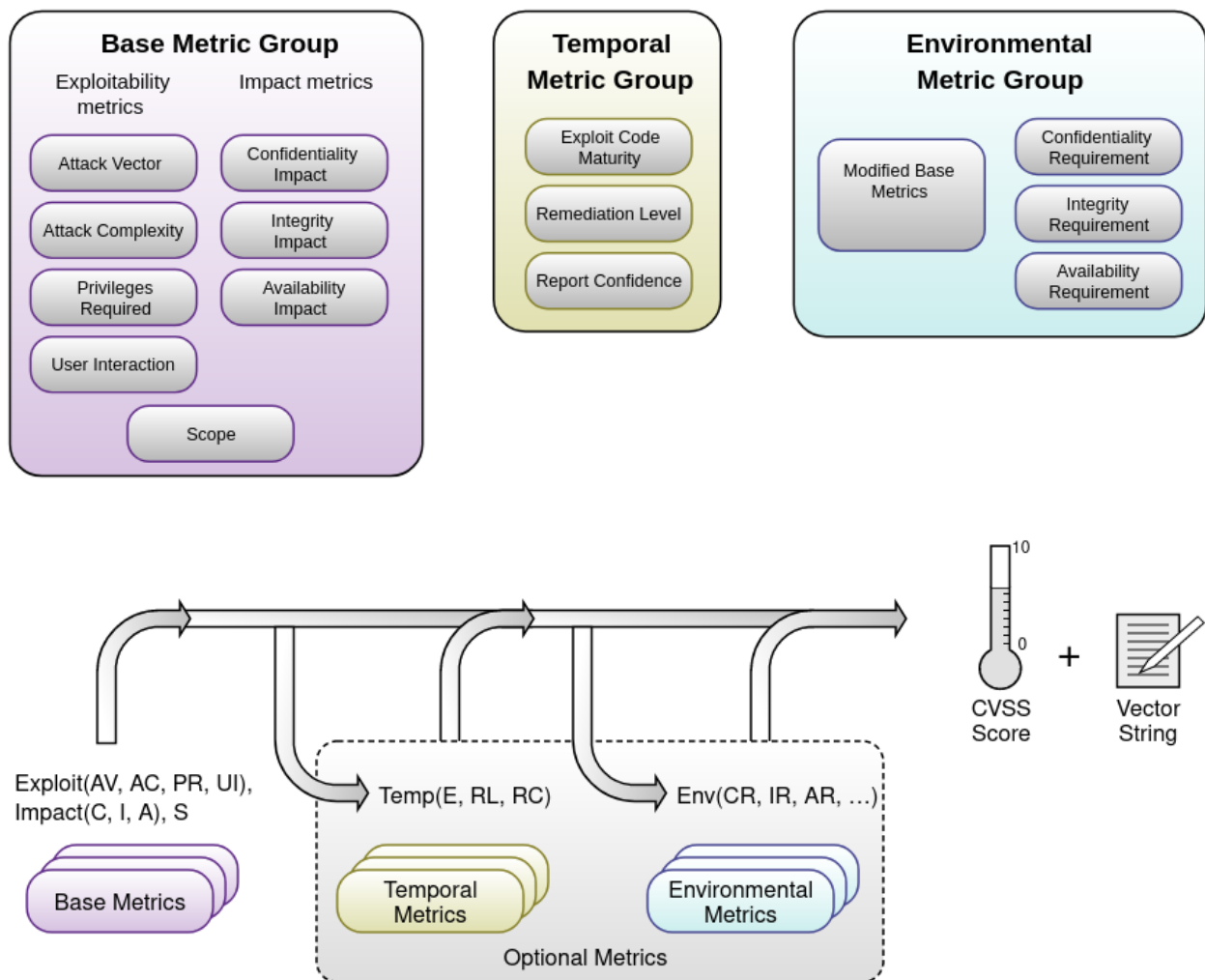
Long-Term: Implement a secure logging and monitoring solution to maintain the ability to view such errors while avoiding the disclosure of them to normal users.

5 Disclaimer and Legal

Penetration tests have inherent risks to the systems under test. Engineers performing testing exercise extreme care to avoid any and all disruption, however, depending on the system and other factors such as traffic, testing environment, and configurations, it is possible that disruptions and in severe cases, complete shutdown of services is possible, though extremely rare. Echo T&E is not liable for any damages, disruptions, or otherwise negative effects of testing on systems and digital infrastructure. It is recommended, when possible, that testing is performed on a representative testing or development environment. Additionally, by agreeing to execution of any testing event, you grant legal authorization to engineers working under Echo T&E to carry out potentially intrusive and exhaustive testing that may appear to be malicious traffic. All testing performed by Echo T&E is considered confidential. Customer information or data will not be shared with anyone or anywhere without written permission of the customer. Echo T&E reserves the right to terminate any services immediately if needed. If, for any reason, testing is terminated prior to the end date agreed, Echo T&E is entitled to the cost of the testing event up to that point.

A Appendix

A.1 CVSS Scoring Diagram



External CVSS Scoring Documentation ([LINK](#))

A.2 Kill Chain

