Cybersecurity

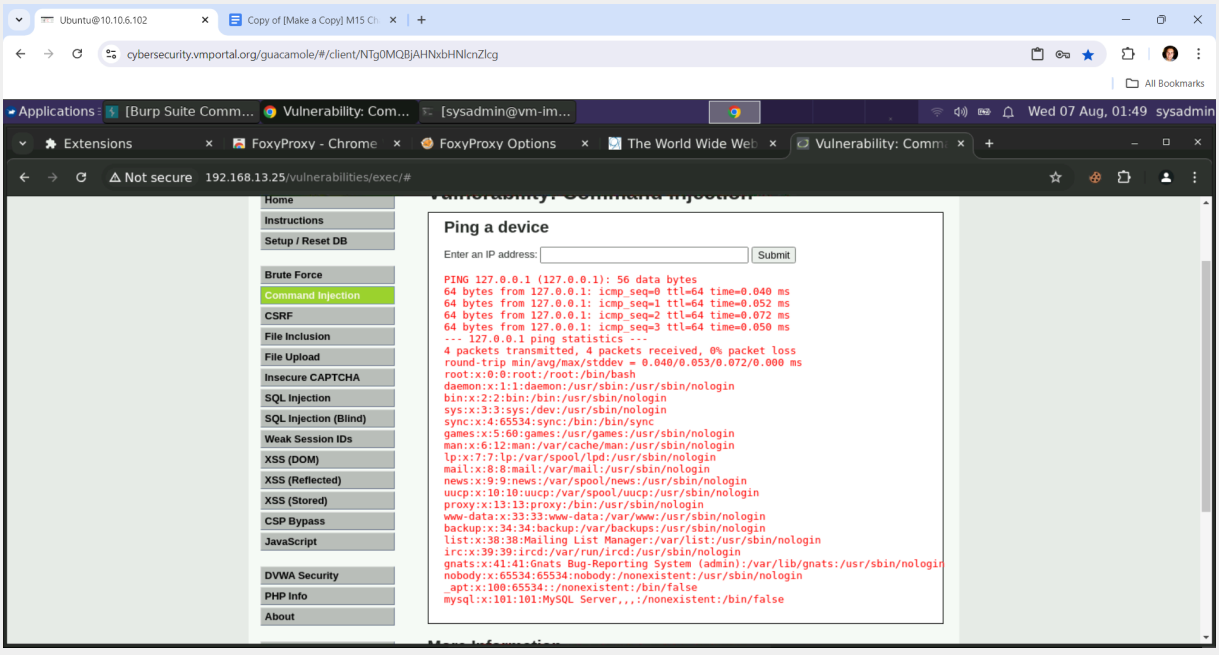
Module 15 Challenge Submission File

Testing Web Applications for Vulnerabilities

Make a copy of this document to work in, and then respond to each question below the prompt. Save and submit this completed file as your Challenge deliverable.

Web Application 1: *Your Wish is My Command Injection*

Provide a screenshot confirming that you successfully completed this exploit:



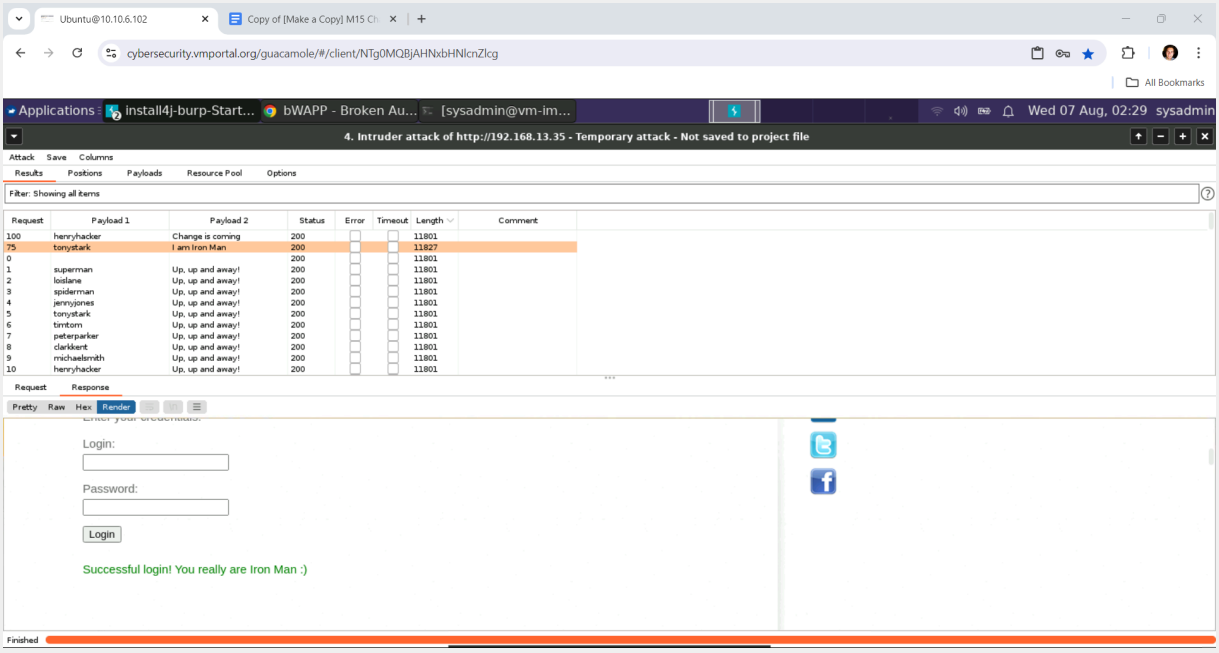
Write two or three sentences outlining mitigation strategies for this vulnerability:

One mitigation strategy that can be applied is input validation code logic to the client and server side code. This ensures that all user inputs are

validated against a strict set of rules. I would also utilize least privilege, which would reduce the impact of this type of attack by limiting system permissions for user services.

Web Application 2: *A Brute Force to Be Reckoned With*

Provide a screenshot confirming that you successfully completed this exploit:

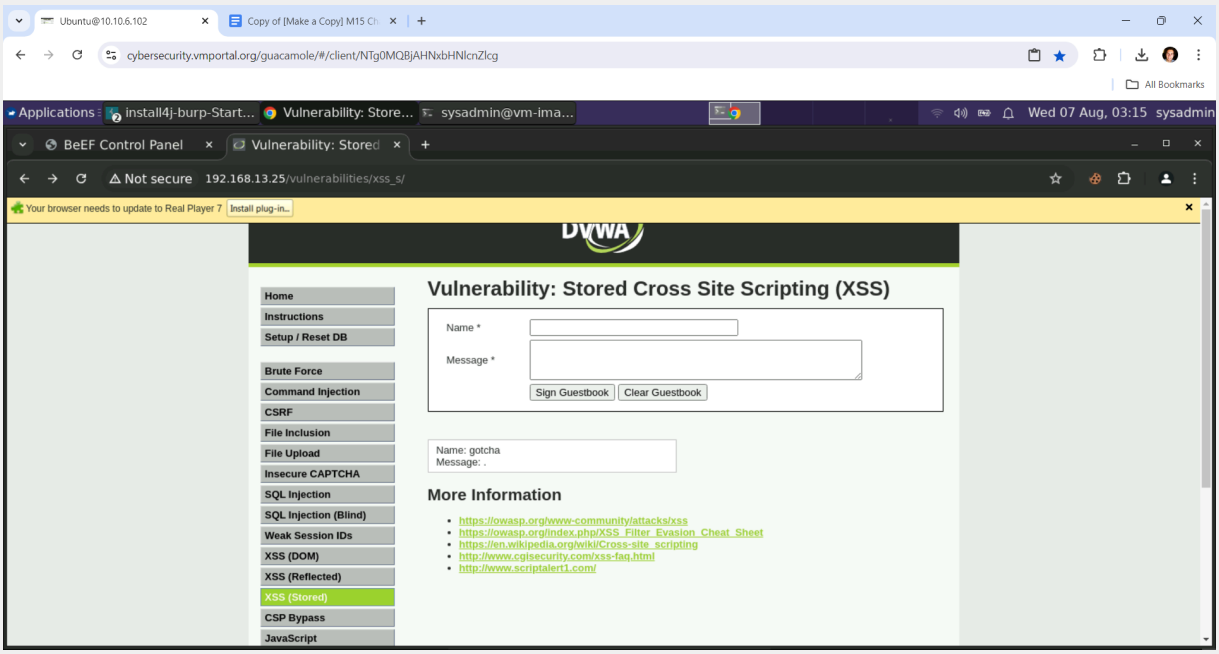


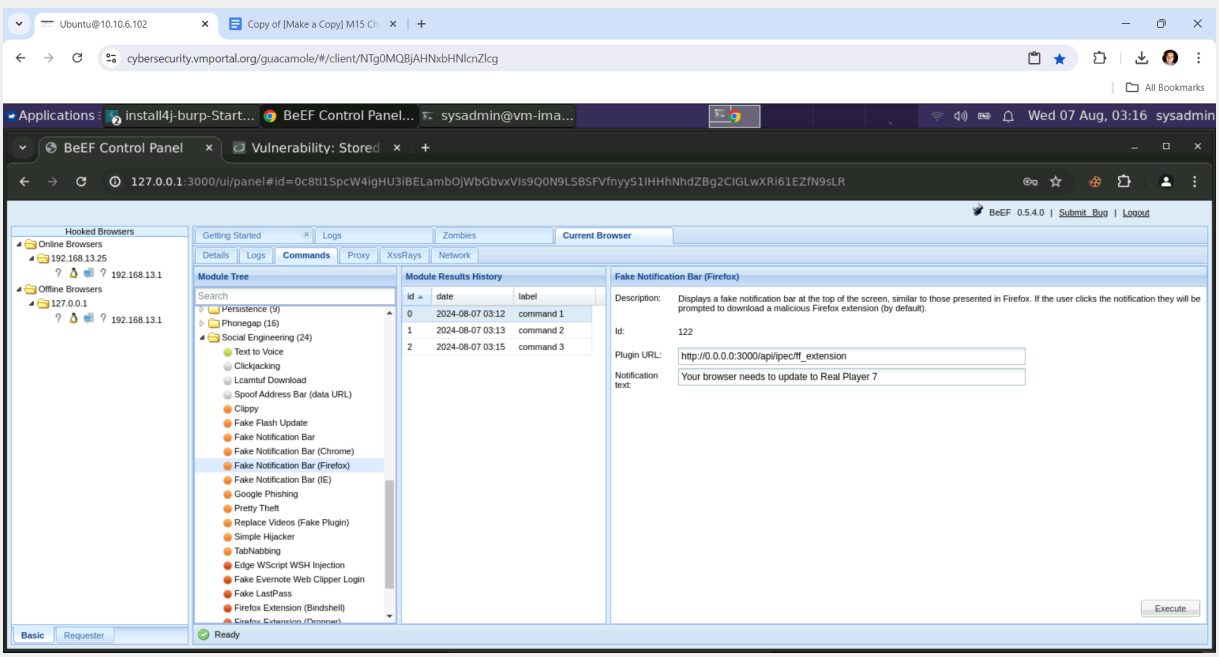
Write two or three sentences outlining mitigation strategies for this vulnerability:

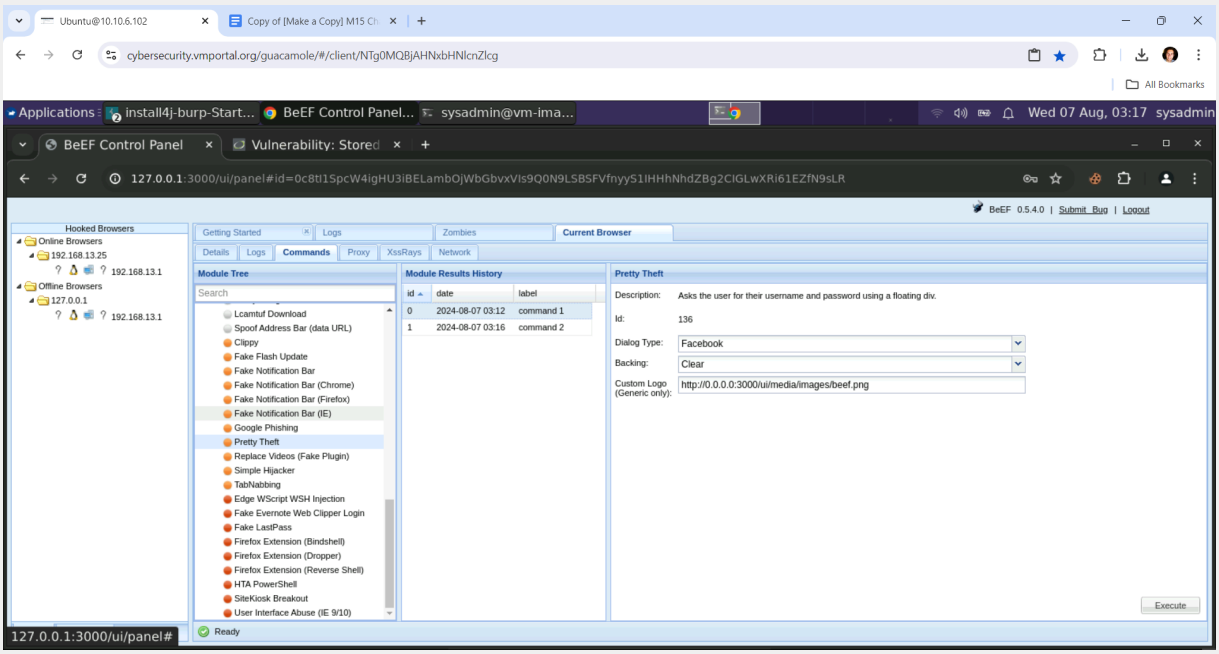
One mitigation strategy for this Brute Force vulnerability would be enforcing strong password policies that use a combination of uppercase, lowercase and special characters of at least 12 characters or more. This would increase the difficulty of a successful brute force attack. Another mitigation strategy would be using Captcha Challenges to verify the identity of the user logging in. This would deter automated bots from using brute force attacks.

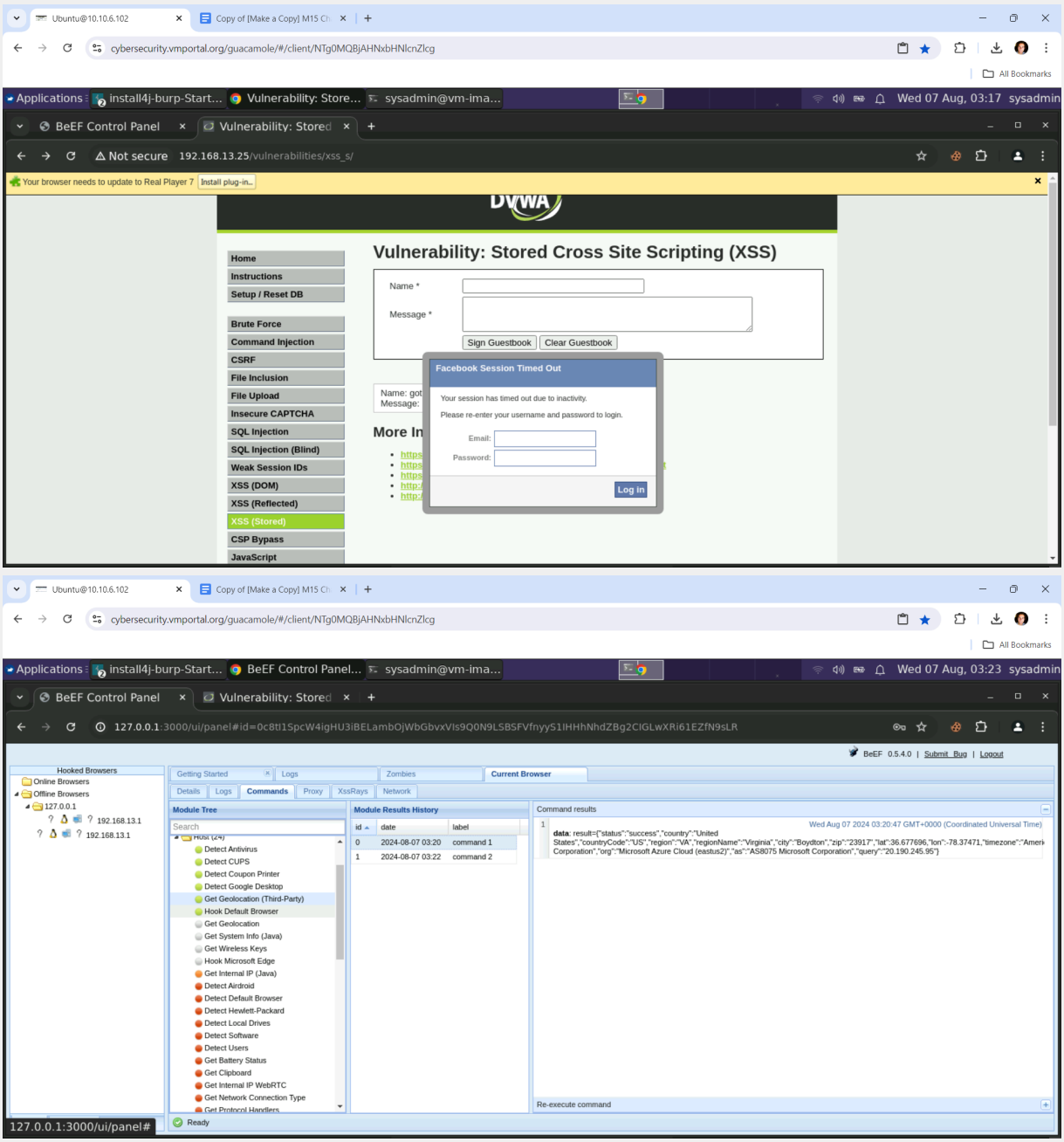
Web Application 3: *Where's the BeEF?*

Provide a screenshot confirming that you successfully completed this exploit:


Ubuntu@10.10.6.102
Copy of [Make a Copy] M15 Ch
cybersecurity.vmportal.org/guacamole/#/client/NTg0MQBjAHNxbHNlcnZlcg
All Bookmarks
Applications
install4j-burp-Start...
Vulnerability: Store...
sysadmin@vm-ima...
Wed 07 Aug, 03:15 sysadmin
BeEF Control Panel
Vulnerability: Stored
Not secure 192.168.13.25/vulnerabilities/xss_s/
Your browser needs to update to Real Player 7
Install plug-in...
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
CSP Bypass
JavaScript
Vulnerability: Stored Cross Site Scripting (XSS)
Name *
Message *
Sign Guestbook
Clear Guestbook
Name: gotcha
Message: .
More Information
https://owasp.org/www-community/attacks/xss
https://owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
https://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html
http://www.scriptalert1.com/


Ubuntu@10.10.6.102
Copy of [Make a Copy] M15 Ch
cybersecurity.vmportal.org/guacamole/#/client/NTg0MQBjAHNxbHNlcnZlcg
All Bookmarks
Applications
install4j-burp-Start...
BeEF Control Panel...
sysadmin@vm-ima...
Wed 07 Aug, 03:16 sysadmin
BeEF Control Panel
Vulnerability: Stored
127.0.0.1:3000/ui/panel#id=0c8tI1SpcW4igHU3iBELambOjWbGbvxVIs9Q0N9LSBSFVfnyyS1IHHhNhdZBg2ClGLwXRi61EZfN9sLR
BeEF 0.5.4.0 | Submit Bug | Logout
Hooked Browsers
Online Browsers
192.168.13.25
192.168.13.1
Offline Browsers
127.0.0.1
192.168.13.1
Getting Started
Logs
Zombies
Current Browser
Details
Logs
Commands
Proxy
XssRays
Network
Module Tree
Search
Persistence (9)
Phonegap (16)
Social Engineering (24)
Text to Voice
Clickjacking
Lcamtuf Download
Spoof Address Bar (data URL)
Clippy
Fake Flash Update
Fake Notification Bar
Fake Notification Bar (Chrome)
Fake Notification Bar (Firefox)
Fake Notification Bar (IE)
Google Phishing
Pretty Theft
Replace Videos (Fake Plugin)
Simple Hijacker
TabNabbing
Edge WScript WSH Injection
Fake Evernote Web Clipper Login
Fake LastPass
Firefox Extension (Bindshell)
Module Results History
id
date
label
0
2024-08-07 03:12
command 1
1
2024-08-07 03:13
command 2
2
2024-08-07 03:15
command 3
Fake Notification Bar (Firefox)
Description:
Displays a fake notification bar at the top of the screen, similar to those presented in Firefox. If the user clicks the notification they will be prompted to download a malicious Firefox extension (by default).
Id:
122
Plugin URL:
http://0.0.0.0:3000/api/ipec/ff_extension
Notification text:
Your browser needs to update to Real Player 7
Execute
Basic
Requester
Ready


Ubuntu@10.10.6.102
Copy of [Make a Copy] M15 Cl
cybersecurity.vmportal.org/guacamole/#/client/NTg0MQBjAHNxbHNlcnZlcg
All Bookmarks
Applications
install4j-burp-Start...
BeEF Control Panel...
sysadmin@vm-ima...
Wed 07 Aug, 03:17 sysadmin
BeEF Control Panel
Vulnerability: Stored
127.0.0.1:3000/ui/panel#id=0c8tI1SpcW4igHU3iBELambOjWbGbvxVIs9Q0N9LSBSFVfnyyS1IHHhNhdZBg2ClGLwXRi61EZfN9sLR
BeEF 0.5.4.0 | Submit Bug | Logout
Hooked Browsers
Online Browsers
192.168.13.25
192.168.13.1
Offline Browsers
127.0.0.1
192.168.13.1
Getting Started
Logs
Zombies
Current Browser
Details
Logs
Commands
Proxy
XssRays
Network
Module Tree
Search
Lcamtuf Download
Spoof Address Bar (data URL)
Clippy
Fake Flash Update
Fake Notification Bar
Fake Notification Bar (Chrome)
Fake Notification Bar (Firefox)
Fake Notification Bar (IE)
Google Phishing
Pretty Theft
Replace Videos (Fake Plugin)
Simple Hijacker
TabNabbing
Edge WScript WSH Injection
Fake Evernote Web Clipper Login
Fake LastPass
Firefox Extension (Bindshell)
Firefox Extension (Dropper)
Firefox Extension (Reverse Shell)
HTA PowerShell
SiteKiosk Breakout
User Interface Abuse (IE 9/10)
Module Results History
id
date
label
0
2024-08-07 03:12
command 1
1
2024-08-07 03:16
command 2
Pretty Theft
Description:
Asks the user for their username and password using a floating div.
Id:
136
Dialog Type:
Facebook
Backing:
Clear
Custom Logo (Generic only):
http://0.0.0.0:3000/ui/media/images/beef.png
Execute
Ready
127.0.0.1:3000/ui/panel#



Write two or three sentences outlining mitigation strategies for this vulnerability:

One mitigation strategy for a stored cross site scripting vulnerability is input sanitation. This removes or encodes possible harmful characters that can be utilized in HTML that threat actors can manipulate, like “&”, “<” and “>.” Another mitigation strategy would be using secure HTTP headers like Content-Security-Policy to restrict sources of executable scripts. This allows scripts to only come from trusted sources.