Introduction to Cybersecurity

David J. Malan
malan@harvard.edu
Securing Software
Phishing
<a href="...">Harvard</a>
Harvard
<a href="https://harvard.edu">Harvard</a>
<a href="https://harvard.edu">https://harvard.edu</a>
https://harvard.edu
<a href="https://harvard.edu">https://harvard.edu</a>
<a href="https://yale.edu">https://harvard.edu</a>
https://harvard.edu

https://yale.edu
Code Injection
Cross-Site Scripting (XSS)
About 6,420,000,000 cats
About 6,420,000,000 cats
www.google.com says

<script>alert('attack')</script>
About 6,420,000,000 cats
About 6,420,000,000 <script>alert('attack')</script></p>
About 6,420,000,000 <script>alert('attack')</script></p>
Google says
attack
About 6,420,000,000 <script>alert('attack');</script>
Reflected
About 6,420,000,000 cats
<a href="...">...</a>
<a href="https://www.google.com/search?q=%3Cscript%3Ealert(%27attack%27)%3C%2Fscript%3E">cats</a>
<script>alert('attack')</script>
alert(document.cookie)
Stored
<script>alert('attack')</script>
mail.google.com says
attack

OK
Adversary
to me

<script>alert('attack')</script>
Character Escapes
About 6,420,000,000

<script>alert('attack')?></script>
About 6,420,000,000 &lt;script&gt;alert('attack')&lt;/script&gt;</p>
About 6,420,000,000 \(<script>alert('attack')</script>\)</p>
&lt; (<)
&gt; (>)
&amp; (&)
&quot; ("")
&apos; (')
...


Content-Security-Policy: script-src https://example.com/
<script src="..."></script>
Content-Security-Policy: style-src https://example.com/
<link href="...
rel="stylesheet">
SQL Injection
SELECT * FROM users
WHERE username = '{username}''
SELECT * FROM users
WHERE username = '{username}'
malan
malan'; DELETE FROM users; --
SELECT * FROM users
WHERE username = '{username}'
SELECT * FROM users
WHERE username = 'malan'; DELETE FROM users; --'}
SELECT * FROM users
WHERE username = 'malan';

DELETE FROM users;
SELECT * FROM users
WHERE username = '{username}' AND password = '{password}';
SELECT * FROM users
WHERE username = '{username}' AND password = '{password}'
SELECT * FROM users
WHERE username = 'malan' AND password = '' OR '1'='1'
SELECT * FROM users
WHERE username = 'malan' AND password = ''
OR '1'='1'
SELECT * FROM users
WHERE (username = 'malan' AND password = '')
OR '1'='1'
SELECT * FROM users
WHERE '1'='1'
Prepared Statements
SELECT * FROM users
WHERE username = '{username}'
SELECT * FROM users
WHERE username = ?
SELECT * FROM users
WHERE username = 'malan' ; DELETE FROM users; --
SELECT * FROM users
WHERE username = '{username}' AND password = '{password}'
SELECT * FROM users
WHERE username = ? AND password = ?
SELECT * FROM users
WHERE username = 'malan' AND password = '' OR '1'='1'
Command Injection
system
eval
Developer Tools
<input disabled type="checkbox">
<input type="checkbox">
Client-Side Validation
<input required type="text">
<input required type="text"/>
<input type="text">
Server-Side Validation
Cross-Site Request Forgery (CSRF)
GET
POST
<form action="https://www.amazon.com/" method="post">
  <input name="dp" type="hidden" value="B07XLQ2FSK">
  <button type="submit">Buy Now</button>
</form>
<form action="https://www.amazon.com/" method="post">
  <input name="dp" type="hidden" value="B07XLQ2FSK">
  <button type="submit">Buy Now</button>
</form>
<form action="https://www.amazon.com/" method="post">
<input name="dp" type="hidden" value="B07XLQ2FSK">
<button type="submit">Buy Now</button>
</form>
<form action="https://www.amazon.com/" method="post">
  <input name="dp" type="hidden" value="B07XLQ2FSK">
  <button type="submit">Buy Now</button>
</form>
<script>
  document.forms[0].submit();
</script>
<form action="https://www.amazon.com/" method="post">
  <input name="dp" type="hidden" value="B07XLQ2FSK">
  <button type="submit">Buy Now</button>
</form>
<script>
  document.forms[0].submit();
</script>
<form action="https://www.amazon.com/" method="post">
    <input name="dp" type="hidden" value="B07XLQ2FSK">
    <button type="submit">Buy Now</button>
</form>
<form action="https://www.amazon.com/" method="post">
    <input name="csrf_token" type="hidden" value="1234abcd">
    <input name="dp" type="hidden" value="B07XLQ2FSK">
    <button type="submit">Buy Now</button>
</form>
<form action="https://www.amazon.com/" method="post">
  <input name="csrf_token" type="hidden" value="1234abcd">
  <input name="dp" type="hidden" value="B07XLQ2FSK">
  <button type="submit">Buy Now</button>
</form>
POST / HTTP/3
Host: amazon.com
X-CSRFToken: 1234abcd
Open Worldwide Application Security Project (OWASP)
Arbitrary Code Execution (ACE)
Remote Code Execution (RCE)
Buffer Overflow
machine code
machine code

return address
machine code

"go to machine code"
machine code

cats
"go to machine code"
machine code

"go to machine code"
machine code
machine code

"go to machine code"
machine code

attack code
"go to machine code"
machine code

attack code

...
machine code

attack code
"go to attack code"
Stack Overflow
Cracking
Reverse Engineering
Malware Analysis
Open-Source Software
Closed-Source Software
App Stores
private key → hash → signature
Package Managers
Operating Systems
Bug Bounty
Common Vulnerabilities and Exposures (CVE)
Common Vulnerability Scoring System (CVSS)
Exploit Prediction Scoring System (EPSS)
Known Exploited Vulnerabilities Catalog (KEV)
Introduction to
Cybersecurity

David J. Malan
malan@harvard.edu