

# Introduction to Cybersecurity

David J. Malan  
malan@harvard.edu

# Securing Software

# Phishing

```
<!DOCTYPE html>
```

```
<html>
```

```
...
```

```
<body>
```

```
...
```

```
</body>
```

```
</html>
```

<p>...</p>

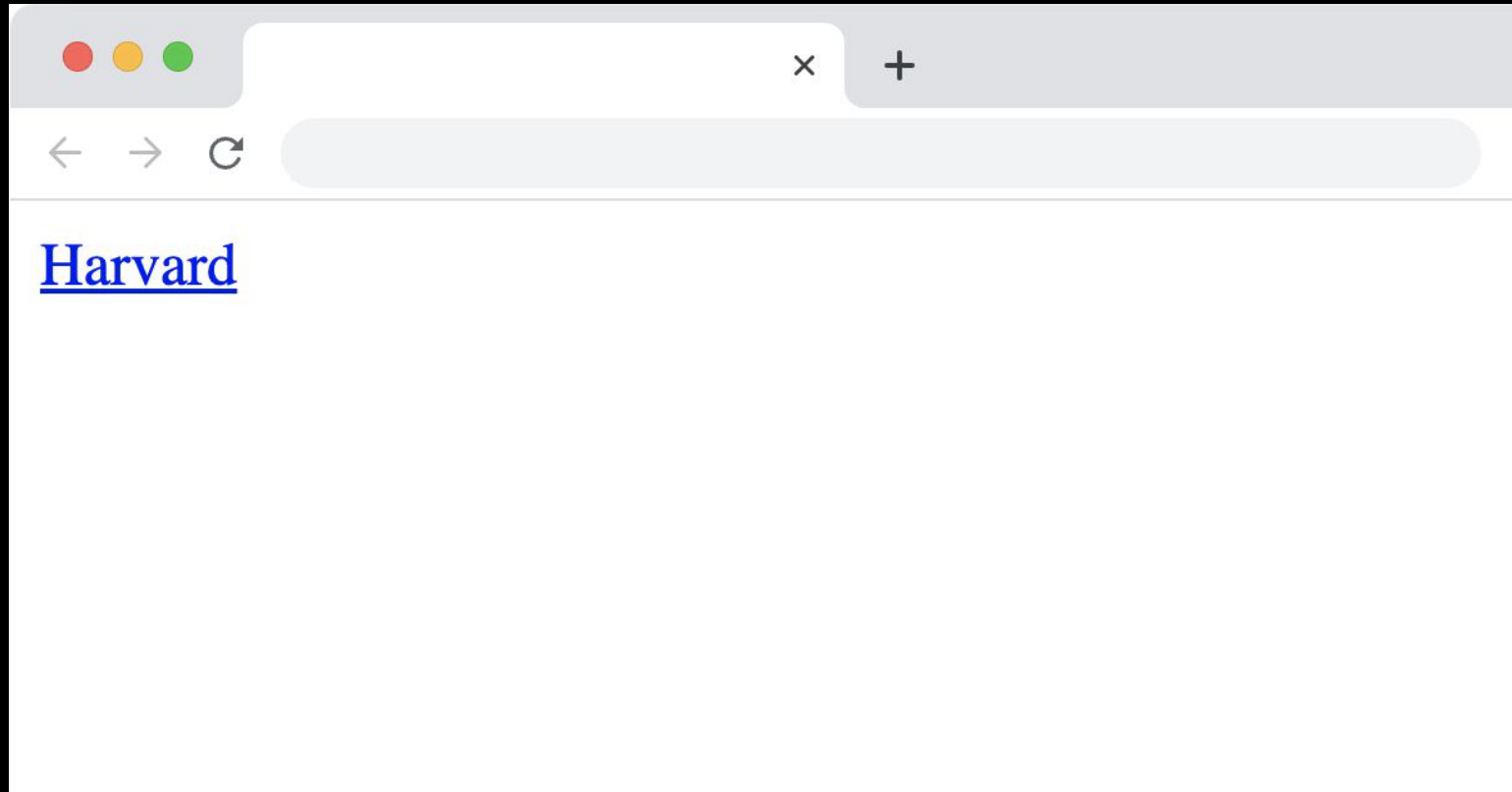
```
<script>...</script>
```

<a>Harvard</a>

```
<a href="...">Harvard</a>
```



```
<a href="https://harvard.edu">Harvard</a>
```





Harvard

https://harvard.edu

```
<a href="https://harvard.edu">Harvard</a>
```

```
<a href="https://harvard.edu">harvard.edu</a>
```



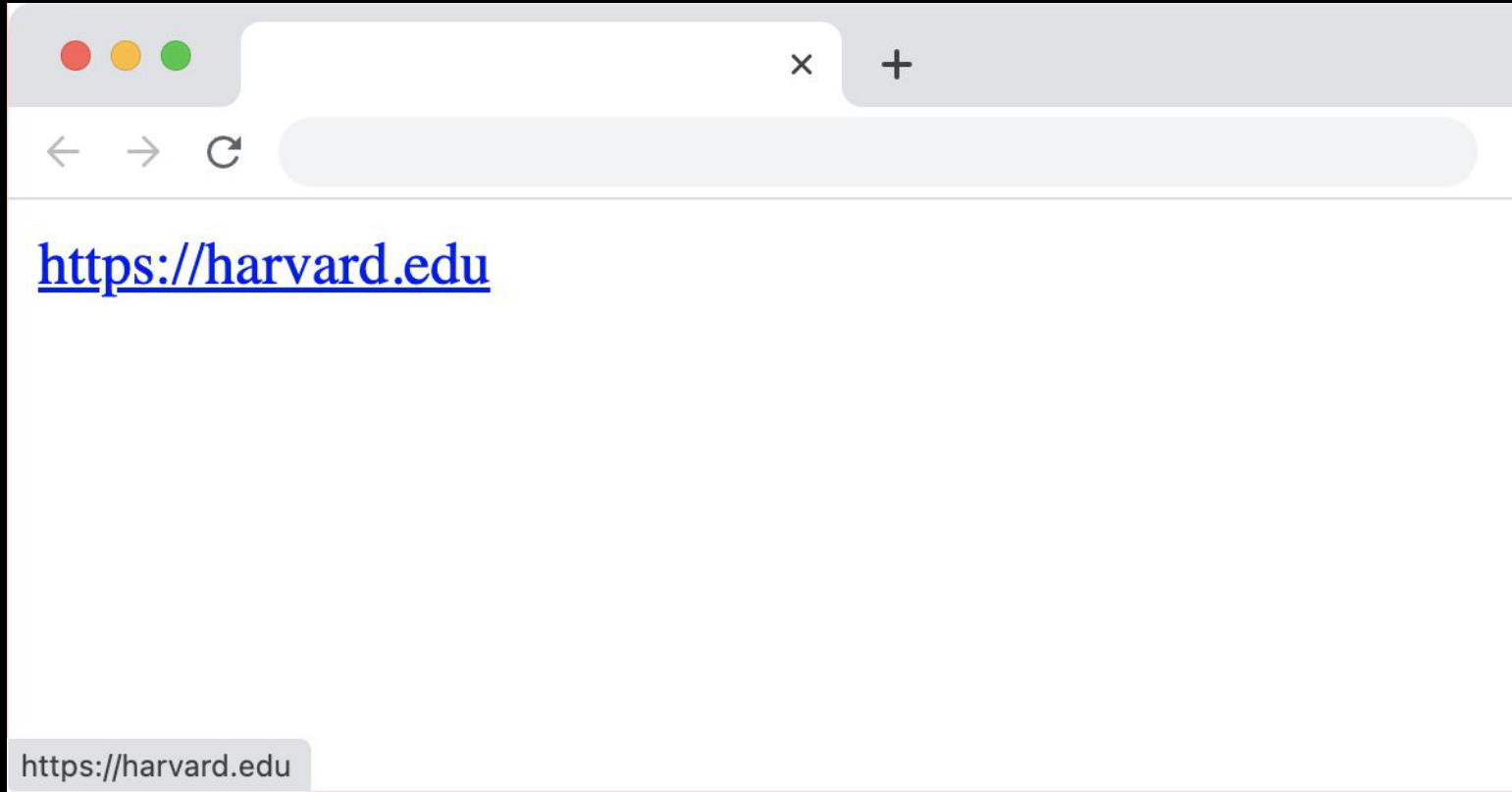
[harvard.edu](https://harvard.edu)

https://harvard.edu

```
<a href="https://harvard.edu">harvard.edu</a>
```

<a href="https://harvard.edu">https://harvard.edu</a>





`<a href="https://harvard.edu">https://harvard.edu</a>`

<a href="https://yale.edu">https://harvard.edu</a>

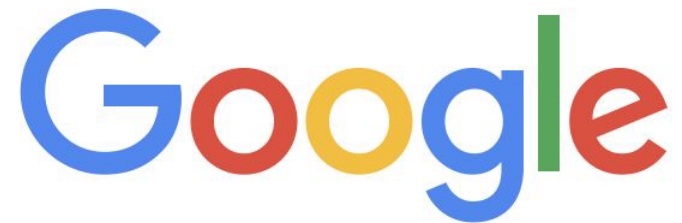


<https://harvard.edu>

https://yale.edu

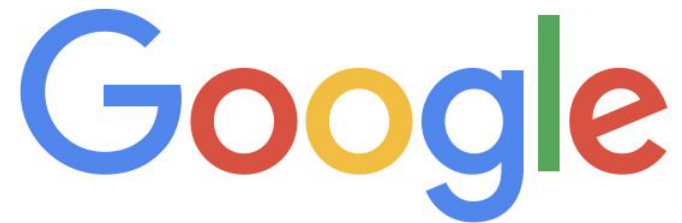
# Code Injection

# Cross-Site Scripting (XSS)



Google Search

I'm Feeling Lucky



🔍 cats|



Google Search

I'm Feeling Lucky





G cats - Google Search



https://www.google.com/search?q=cats



cats



About 6,420,000,000 cats

<p>About 6,420,000,000 cats</p>



G cats - Google Search



https://www.google.com/search?q=cats



cats



About 6,420,000,000 cats



G cats - Google Search



https://www.google.com/search?q=cats



`<script>alert('attack')</script>`



About 6,420,000,000 cats



cats - Google Search



https://www.google.com/search?q=<script>alert%28%27attack%27%29<%2Fscript>



<script>a attack



OK

About 6,420,000,000

<p>About 6,420,000,000 cats</p>

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

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





cats - Google Search



https://www.google.com/search?q=<script>alert%28%27attack%27%29<%2Fscript>



<script>a attack



OK

About 6,420,000,000



<script>alert('attack');</script> x



https://www.google.com/search?q=<script>alert%28%27attack%27%29%3B<%2Fscript>



<script>alert('attack');</script>



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

Reflected

```
<a href="...">...</a>
```



G cats - Google Search



https://www.google.com/search?q=cats



cats



About 6,420,000,000 cats

```
<a href="...">...</a>
```

```
<a href="https://www.google.com/search?q=cats">cats</a>
```

```
<a href="https://www.google.com/search?q=cats">cats</a>
```



```
<a href="https://www.google.com/search?q=cats">cats</a>
```

<a href="https://www.google.com/search?q=%3Cscript%3Ealert%28%27attack%27%29%3C%2Fscript%3E">cats</a>

```
<script>alert('attack')</script>
```

```
<script>alert(document.cookie)</script>
```

Stored

## New Message



Recipients

---

Subject

---

`<script>alert('attack')</script>`

Send





Gmail



https://mail.google.com



mail.google.com says

attack

OK

# Adversary

to me ▼

```
<script>alert('attack')</script>
```



# Character Escapes

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

<p>About 6,420,000,000 &lt;script>alert('attack')&lt;/script></p>

<p>About 6,420,000,000 &lt;script>alert('attack')&lt;/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}'
```

malan

' 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 (username = 'malan' AND password = '')  
OR '1'='1'
```



```
SELECT * FROM users  
WHERE '1'='1'
```

# Prepared Statements

1

”

```
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 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



[Buy Now](https://www.amazon.com/dp/B07XLQ2FSK)

[Buy Now](https://www.amazon.com/dp/B07XLQ2FSK)

```

```

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)

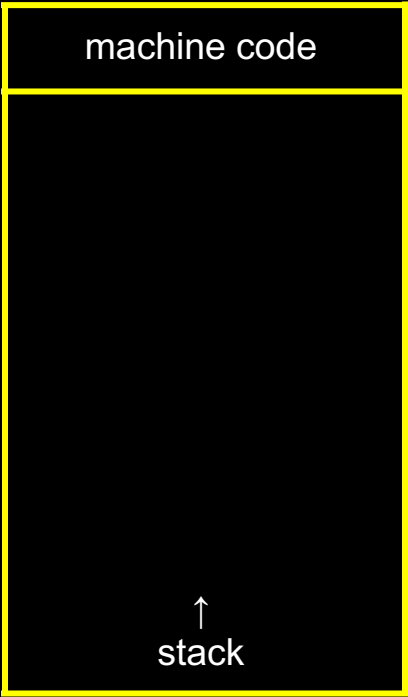
# Arbitrary Code Execution (ACE)

# Remote Code Execution (RCE)



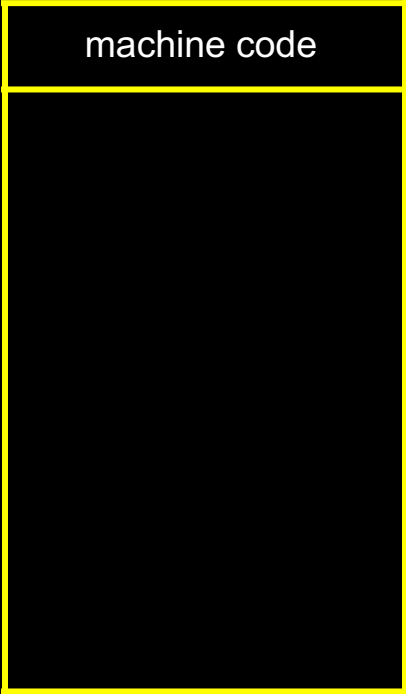
# Buffer Overflow

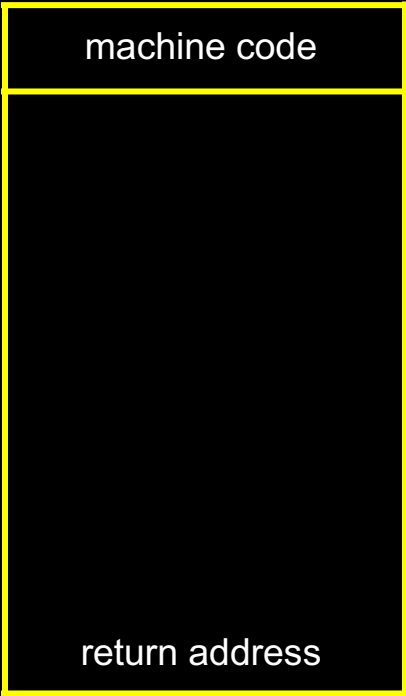
machine code

A diagram of memory layout on a black background. It consists of a large yellow-outlined rectangle. The top portion of this rectangle is a smaller rectangle labeled "machine code". The bottom portion of the large rectangle is empty, with an upward-pointing arrow and the word "stack" at the very bottom, indicating the stack grows upwards into this space.

↑  
stack

machine code

A diagram of a memory block. It consists of a yellow rectangular border. The top portion of the rectangle is a smaller rectangle containing the text "machine code" in white. The bottom portion of the rectangle is empty.



machine code

return address

machine code

"go to machine code"

machine code

cats  
"go to machine code"

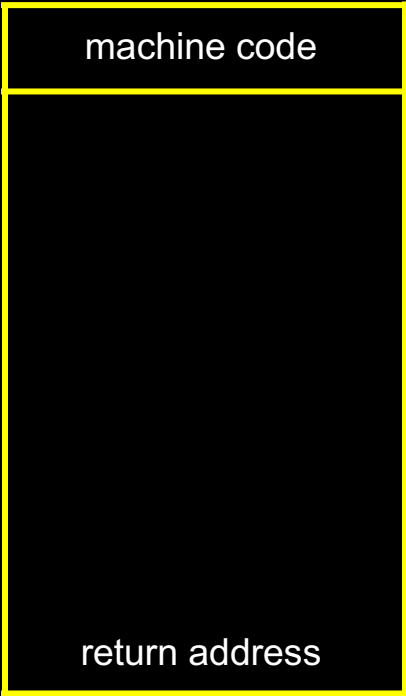
machine code

"go to machine code"

machine code

A diagram of a memory block. It consists of a yellow rectangular border. The top portion of the rectangle is a smaller rectangle containing the text 'machine code' in white. The bottom portion of the rectangle is empty.





machine code

return address

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



A diagram of memory layout. It consists of a large yellow rectangle divided into two horizontal sections. The top section is labeled 'machine code'. The bottom section is mostly empty, with an upward-pointing arrow and the word 'stack' at the bottom center, indicating the stack grows upwards from the bottom of this section.

machine code

↑  
stack

# Cracking



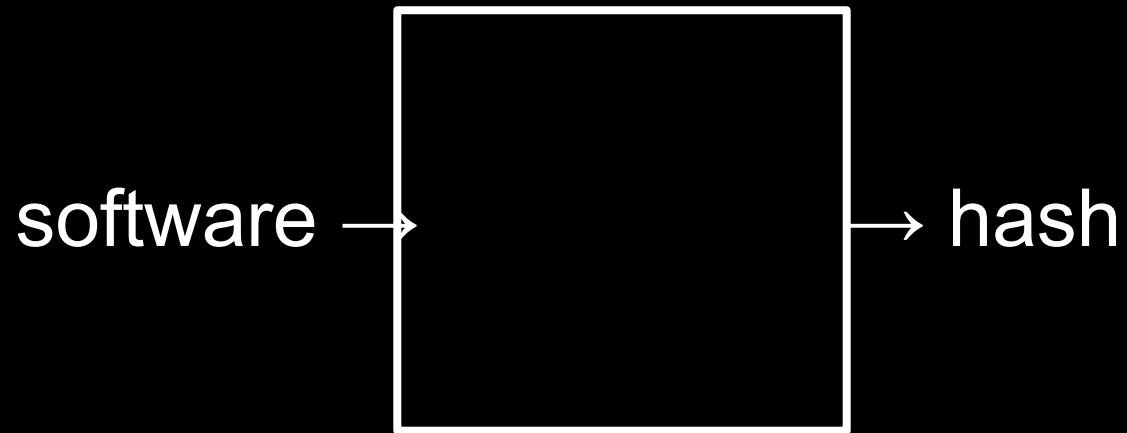
# Reverse Engineering

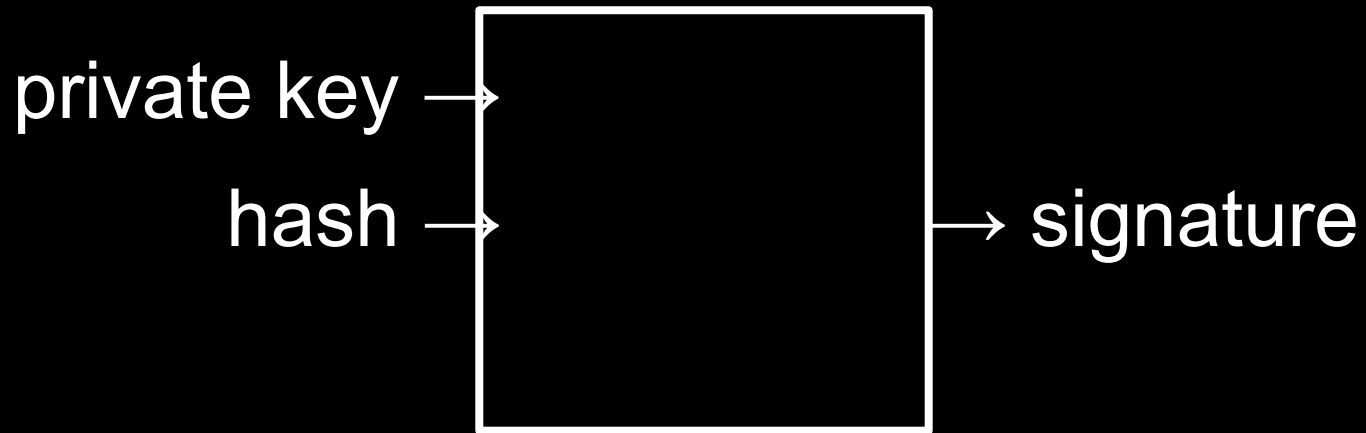
# Malware Analysis

# Open-Source Software

# Closed-Source Software

# App Stores





# 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