Cracking ShadowCrypt: The limitations of web privacy

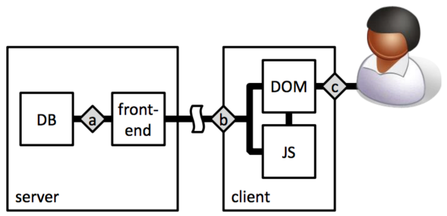
Michael Freyberger[1], Prateek Mittal[1]
[1]*Electrical Engineering, Princeton University*

MOTIVATION

- Users are forced to trust web applications with a lot of personal information

http://network-telephone-services.com/wp-content/uploads/2014/04/network-telephone-services-calendar.jpg

PROBLEM OVERVIEW



Warren He, Devdatta Akhawe, Sumeet Jain, Elaine Shi, and Dawn Song. 2014. ShadowCrypt: Encrypted Web Applications for Everyone. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (CCS '14).

Definitions

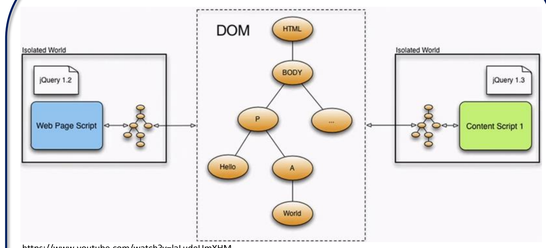
- **server** computer associated with an IP address that delivers web pages to users
  - **DB** component that stores user data
  - **front-end** logic on the server to handle user requests, query the database, and deliver HTML/JS/CSS to the client over the network
- **client** web browser that executes the JS and transforms the HTML into the DOM
  - **DOM** tree structured representation of the user interface
  - **JS** code that is executed in the browser

Chokepoints

The block diagram above highlights three chokepoints (a, b, and c)**.** A security system executing at a particular **chokepoint** ensures all components to the left of the chokepoint only have access to encrypted data, and are part of the trusted computing base. Components to the right of the chokepoint have access to the plaintext.

BROWSER PRIMATIVES

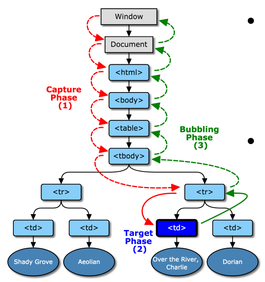
Isolated World



https://www.youtube.com/watch?v=laLudeUmXHM

- Chrome Extensions have full access to the web pages DOM
- Extensions run in an isolated JS environment
- Extensions have access to privileged APIS and execute at a higher privilege level

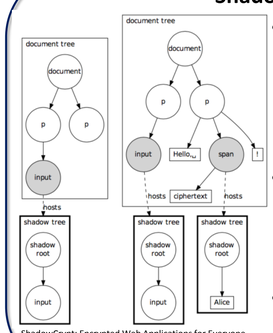
JavaScript Events



http://www.w3.org/TR/DOM-Level-3-Events/#event-flow

- Events can be registered to execute during the Capture phase or during the Bubbling phase.
- Event propagation can be stopped at any point during the propagation path by one of the event handlers

Shadow DOM



ShadowCrypt: Encrypted Web Applications for Everyone.

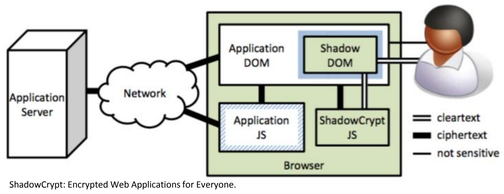
- Provides the ability to create a separate encapsulated DOM attached to an existing element
- Helps programmers avoid breaking sites due to conflicting CSS selectors or JS variables
- Each element can host at most one shadow DOM

Mutation Observer

- API for developers to register callbacks upon changes to the DOM
- Registered onto a specific target
- Can observe target's child list, attributes, and subtree

ShadowCrypt

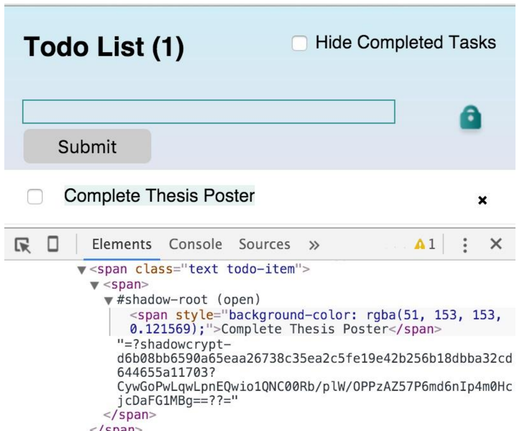
Architecture



ShadowCrypt: Encrypted Web Applications for Everyone.

- ShadowCrypt JS runs before any web page script in a secure environment.
- The user reads and writes all sensitive information through the Shadow DOM, which is created by ShadowCrypt JS

Observations



- ShadowCrypt has a clean UI which makes this extension very useable
- It clearly highlights data which has been stored on the server as encrypted data, and then decrypted on the client

Attacks

User Interface Attack

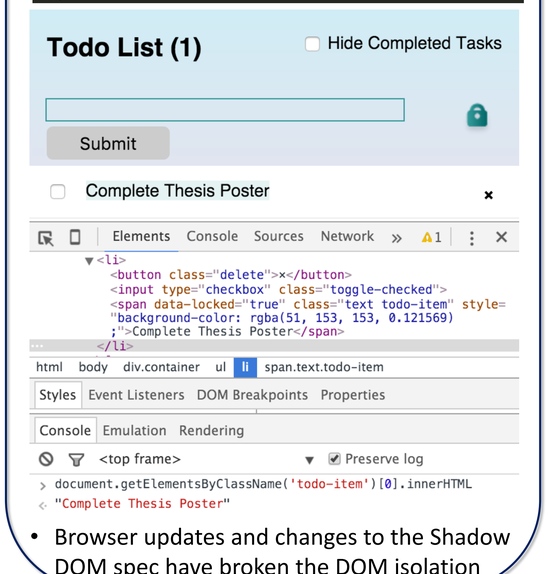
```
<form class="new-task">
  <input class="main-input" type="checkbox" name="text" />
  <input class="false-input" type="input" name="false-text" />
  <input type="submit" value="Submit">
</form>
```

```
//Run the attack when the DOM is rendered fully.
Template.body.rendered = function() {
  // select the target node
  var target = document.getElementsByClassName('main-input')[0];
  target.type = "input";
};
```

- Upon page render an insecure checkbox element can be set to type input without ShadowCrypt reacting to the change
- The insecure input node can be rendered on top of a secure node tricking the user into believing the data is being entered securely

Shadow DOM Boundary Attack

```
var inputNode = event.target.getElementsByClassName('main-input')[0];
var text = '';
if (Attacker.isLocked(inputNode)) {
  text = event.target.getElementsByClassName('main-input')[0].shadowRoot.
    childNodes[1].childNodes[0].value;
  var locked = true;
}
else {
  var text = event.target.getElementsByClassName('main-input')[0].value;
  var locked = false;
}
Tasks.insert({
  text: text,
  locked: locked,
  createdAt: new Date()
});
```



- Browser updates and changes to the Shadow DOM spec have broken the DOM isolation

Conclusions and Future Work

- Currently ShadowCrypt does not provide an isolated DOM allowing the client access to the plaintext
- Browsers are constantly updating and relying upon a moving target is difficult
- Perform a user study to gain insight into the effectiveness of the User Interface attack
- Look into countermeasures in order to provide a security system at chokepoint C
- Research security systems that run at chokepoint B