
milcom
Military Communications for the 21st Century
November 12–14, 2019 • Norfolk, VA, USA
Defining Multi-Domain Command and Control

milcom
Military Communications for the 21st Century
November 12–14, 2019 • Norfolk, VA, USA
Defining Multi-Domain Command and Control

An Intelligent System for Preventing SSL Stripping-based Session Hijacking Attacks

Mainuddin Ahmad Jonas, Md. Shohrab Hossain, Risul Islam, **Husnu S. Narman**, Mohammed Atiquzzaman

Outlines

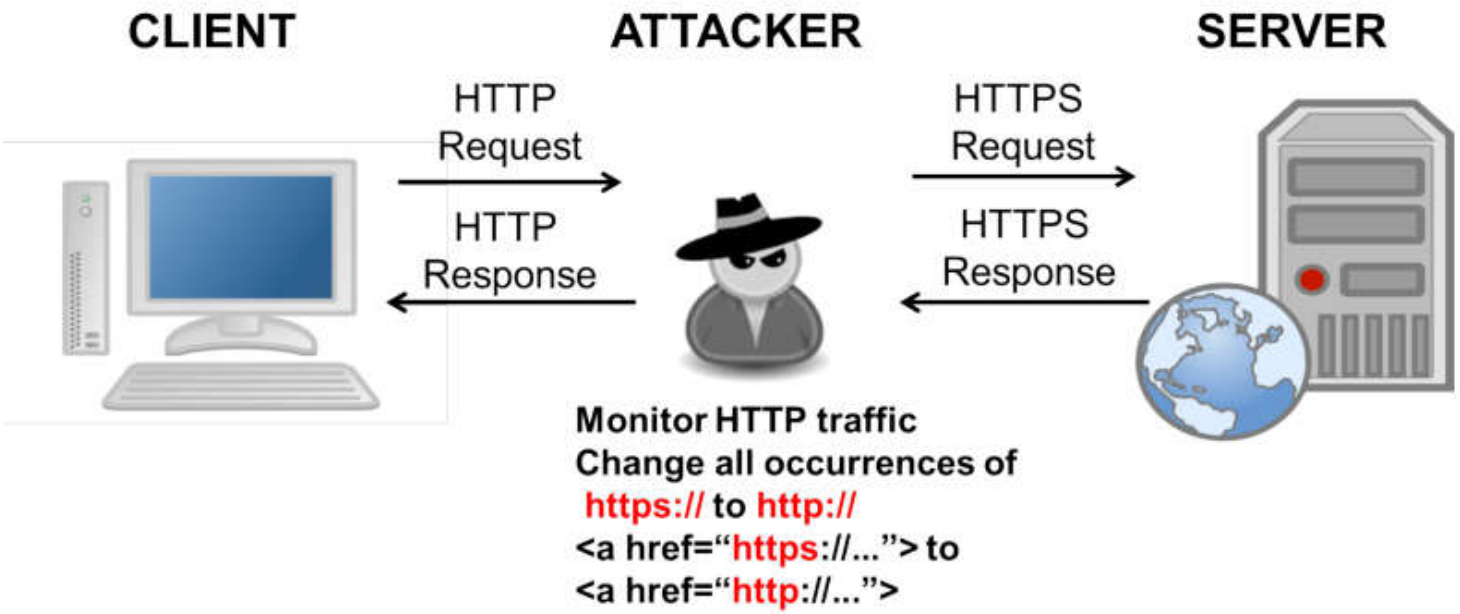
- Motivation
- Problem
- Contributions
- Results
- Conclusion

Identifying Jargons

An Intelligent System for Preventing SSL Stripping-based Session Hijacking Attacks

- SSL Stripping
- SESSION HIJACKING

Motivation
SSL Stripping Attacks



Motivation

- SSL consists of three protocols
  - Handshake Protocol, Record Protocol, Alert Protocol.
- Handshake protocol
  - Establishes a secure connection between the server and the client
- Alert protocol
  - Custom messages whenever an intrusion is detected
- The handshake protocol
  - The most vulnerable part of the SSL connection
  - Done over unencrypted plain text

Problem

- Attacks on SSL: two types primarily
  - SSL Sniffing attacks
    - Spoofed certificates
    - Browsers show warnings
  - SSL Stripping attacks.
    - SSL Stripping type of attacks does not result in any warning messages for users, making them more dangerous.

Some Solutions

- Hashed the password sent by the client with the server's certificate
- Hproxy: It built a profile of safe SSL-enabled websites from the history of requests and responses.
- SSLock: enforcing special protected domains which enforce SSL connection.
- HTTPSLock: enforcing the HTTPS protection and forbid users to embrace invalid certificates.
- ISAN HTTPS Enforcer: handling redirections from the client side and overcoming the problem of user bypassing security warnings

Problems which are not well addressed

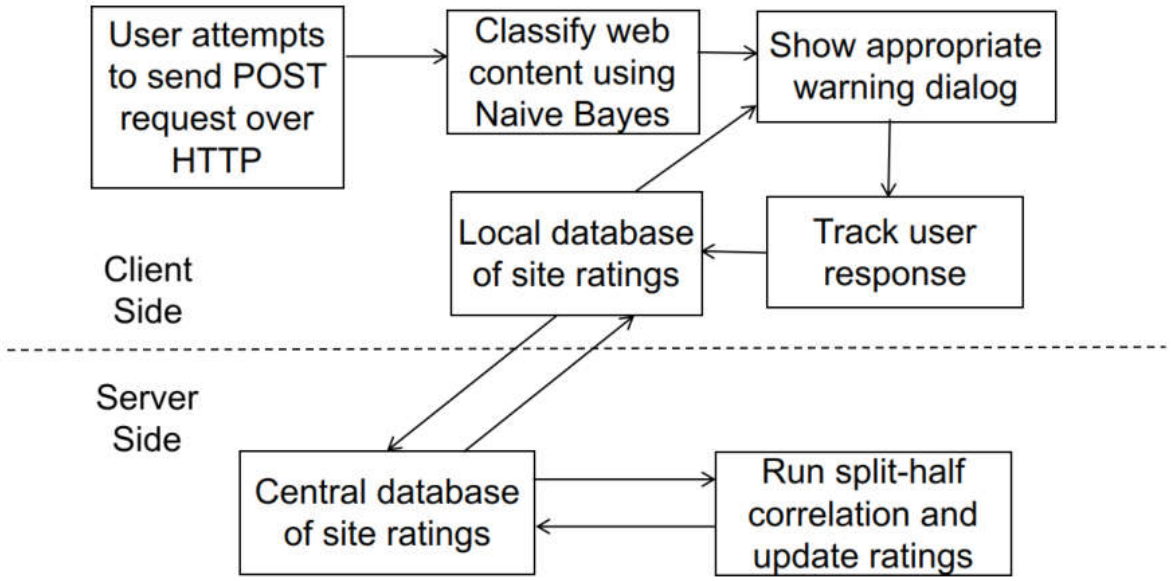
- User behavior towards security issues
  - SSL stripping is successful primarily because users are not educated about the difference between HTTP and HTTPS connections, and therefore are not aware of the importance of using encrypted connections while sending sensitive data to websites.
  - Users cannot to be expected to type in HTTPS in the URL bar to ensure secure a connection.
  - Users have a habit of ignoring warning dialogs even if the warning cautions against the possibility of leakage of sensitive data.
  - False negative rate is very high, while the false positive rate is relatively low in user response towards security warnings.

Contributions

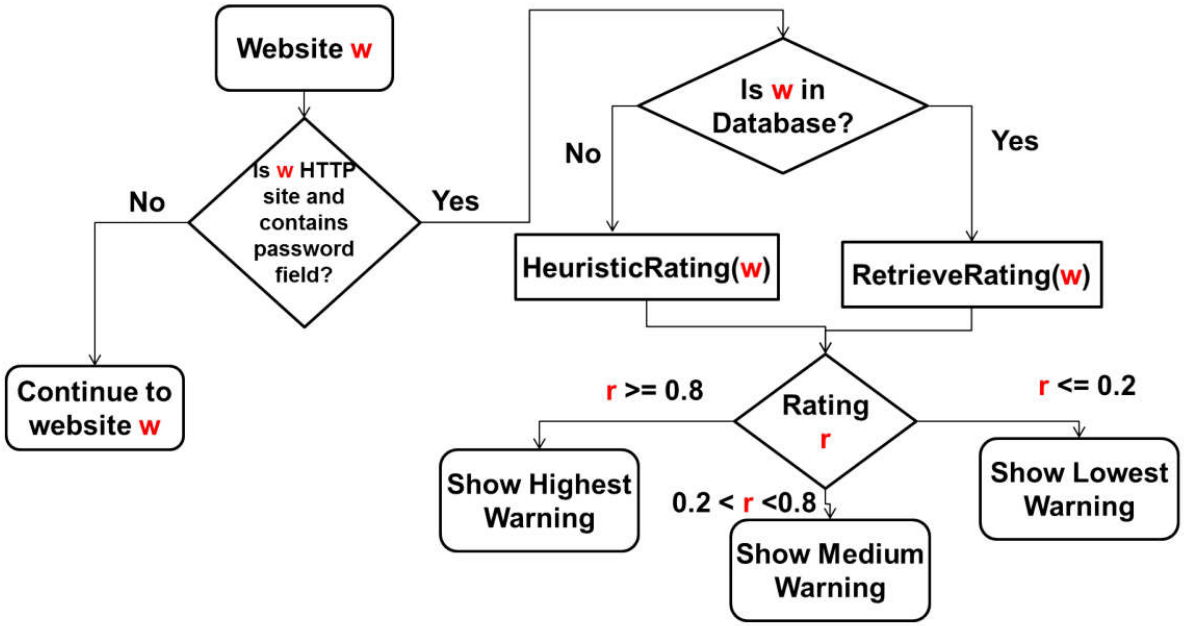
- An intelligent system to prevent SSL Stripping based session hijacking attacks
- The system is designed to strike a delicate balance between security and user friendliness.

Proposed Features

- Client-Side
  - Local Database
  - Rating
  - Warning System
- Server Side
  - Data gathering from Users
  - Classification
  - Rating Update



Client Side

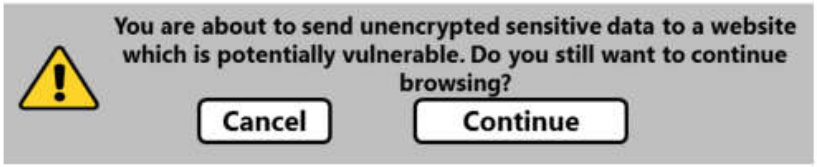


Class of website	Rating	Security Level
Banking	1.0	High
E-commerce	0.9	High
Education/Email	0.7	Medium
Social Media	0.5	Medium
Miscellaneous	0.1	Low

Client-Side Warning

- Highest:

- Medium:



- Lowest

Split-half correlation algorithm

Algorithm 1 To measure the point of 50% regression

Input: Websites in database

Output:t point of 50% regression, t = (1-r)/r * x

1: get all websites from database
2: partition all websites into two sets of equal size
3: run a correlation between the two sets
4: find correlation coefficient r
5: x = average sample size
6: point of 50% regression, $t = (1 - r)/r * x$
7: **return** t

Ratings update algorithm

Algorithm 2 To update the rating of each website

Input: Websites and entries in database with t

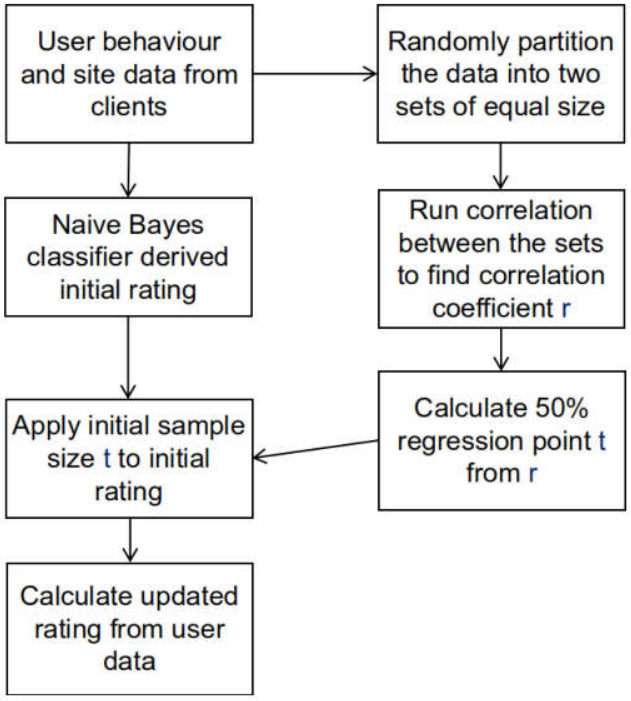
Initialization :

```
for each website w in database do
    sample of w = t
    sum of w = initialRating(w) * t
    score of w = sum of w / sample of w
end for
```

Updating :

```
for each entry d in database do
    w = get website of d
    p = warning level of d for w
    a = user_action of d for w
    if a == respect then
        increment sample of w by 1
        increment sum of w by 1
        score of w = sum of w / sample of w
    else if a == bypass then
        if p == high then
            increment sample of w by 0.8
            score of w = sum of w / sample of w
        else if p == medium then
            increment sample of w by 0.5
            score of w = sum of w / sample of w
        else if p == low then
            increment sample of w by 0.2
            score of w = sum of w / sample of w
        end if
    end if
end for
```

Server Side



$$rating_{new} = \frac{rating_{old} \times n_{sample} + n_{accept}}{n_{sample} + n_{accept} + weight \times n_{reject}}$$

- Weight depends on the current warning level of the website
  - For high level: weight = 0.8
  - For medium level: weight = 0.5
  - For low level: weight = 0.2

Results

- Tools and Samples
- User Behavior Simulation
- Rating Update

Tools and Samples

- A sample of 100 websites of different categories used to train the initial Naïve Bayes classifier
- 5 websites were used for simulating user behavior
- *Squid* proxy software on Ubuntu used to filter and redirect traffic
- *w3m* UNIX tool used to extract text from websites

User Behavior Simulation

Website	Class	Security Level	Rating
dutchbanglabank.com	Banking	High	1.0
gmail.com	Education/Email	Medium	0.7
buet.ac.bd	Education/Email	Medium	0.7
facebook.com	Social Media	Medium	0.5
stackoverflow.com	Miscellaneous	Low	0.1

Initial Rating

Collected User Behaviors in Server Side

Website	Accepted Warning	Rejected Warning
dutchbanglabank.com	156	44
gmail.com	122	78
buet.ac.bd	147	53
facebook.com	68	132
stackoverflow.com	11	189

Splitting and Correlation

Website	Sample 1 Accept %	Sample 2 Accept %
dutchbanglabank.com	70	86
gmail.com	69	53
buet.ac.bd	78	69
facebook.com	40	28
stackoverflow.com	5	6

- Split-half Correlation Technique (Cronbach's Alpha can be better to reduce errors but computationally more expensive)
- Correlation coefficient, r = 0.916
- t = (1 – 0.916) / 0.916 * 100 = 9.17
- So 9 samples is the point of 50% regression.

Rating Update

Website	Updated rating	Security Level
dutchbanglabank.com	$r = \frac{9*1.0+156}{9+156+0.8*44} = 0.82$	High
gmail.com	$r = \frac{9*0.7+122}{9+122+0.5*78} = 0.75$	Medium
buet.ac.bd	$r = \frac{9*0.7+147}{9+147+0.5*53} = 0.84$	High
facebook.com	$r = \frac{9*0.5+68}{9+68+0.5*132} = 0.51$	Medium
stackoverflow.com	$r = \frac{9*0.1+11}{9+11+0.2*189} = 0.21$	Medium

After Update

Initial Rating

Website	Class	Security Level	Rating
dutchbanglabank.com	Banking	High	1.0
gmail.com	Education/Email	Medium	0.7
buet.ac.bd	Education/Email	Medium	0.7
facebook.com	Social Media	Medium	0.5
stackoverflow.com	Miscellaneous	Low	0.1

Real-world Implementation

- Can be integrated into the browser, or be provided as an extension
- Could be a system-wide app for smartphone devices
- Ensuring privacy would be critical, so all communication between client and server should be encrypted
- No personally identifiable information is required from the client, and hence should not be collected by server

Preventing Adversarial Attacks

- Potential adversaries may try to poison the integrity of our database
- One solution is to block bulk requests from suspicious IP addresses
- Another is to require users to register.
- User verification can be done once a week or month.

Conclusions

- Security is more of a human problem, than a technical problem
- Human behavior should be the most important factor in security solutions
- User feedback is core part of our model and is used directly in the algorithms
- This model could be applied to other tasks, for example, App Store reviews, content moderation on social media etc.

Thank You
Questions

Husnu Narman
narman@marshall.edu
https://hsnarman.github.io/