Talking with Familiar Strangers: An Empirical Study on HTTPS Context Confusion Attacks

Original research by:

Mingming Zhang, Xiaofeng Zheng, Kaiwen Shen, Ziqiao Kong, Chaoyi Lu,Yu Wang, Haixin Duan, Shuang Hao, Baojun Liu and Min Yang

Tsinghua university

Presented by:

Allen Zhao

Introduction

- Google states that over 93% of its webpages are using HTTPS. Due to this vast amount of usage, it could be a disaster if attackers find ways to intercept the client-server connections in HTTPS.

- Attackers have nowadays have turned their goal to smash the end-to-end protection of the HTTP protocol using man-in-the-middle attacks.

- Common MITM attacks include, SSL sniffing, SSL/TLS stripping, etc.

- Today's topic: SCC attack(HTTP**S** **C**ontext **C**onfusion attack).

Sniffing attack

- Attacker creates SSL connection with both the client side and server side
- Pretend to be the client at server side.
- Pretend to be the server at client side.
- Therefore the attacker can “sniff” the data that is transmitted from both sides.

Stripping attack

- Attacker incercepts the initial HTTP connection
- Replace the secure links(e.g from HTTPS to HTTP) returned by the server with plaintext ones
- Downgrade the SSL channel

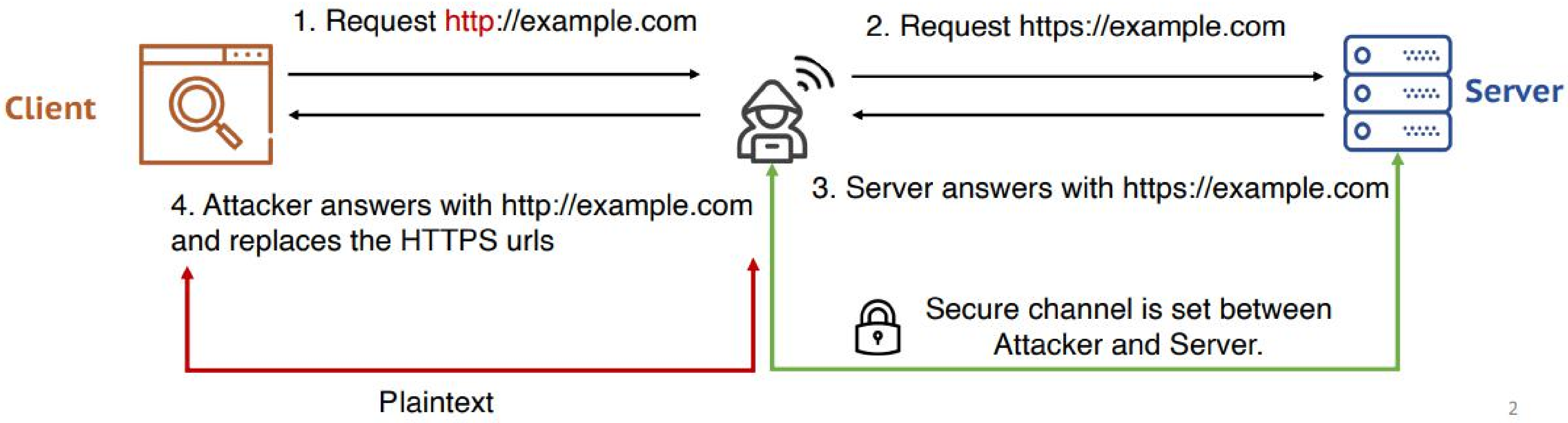
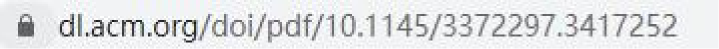

1. Request http://example.com
2. Request https://example.com
Client
Server
3. Server answers with https://example.com
4. Attacker answers with http://example.com and replaces the HTTPS urls
Secure channel is set between Attacker and Server.
Plaintext

Image from Dr. Mingming Zhang presentation slides, link: https://shenkaiwen.com/files/papers/SCC_CCS20_pre.pdf

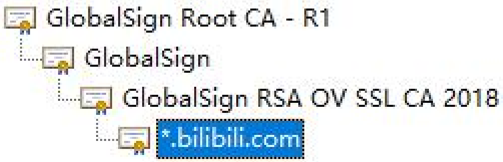
Difficulties of these attacks

Sniffing based attacks requires the adversary to install the root certificate at client side while stripping-based attacks requires interception of initial HTTPs connection and communicate with client-side in plaintext. In fact during the overall course of the attack the client is not protected by any legal and active certificate.

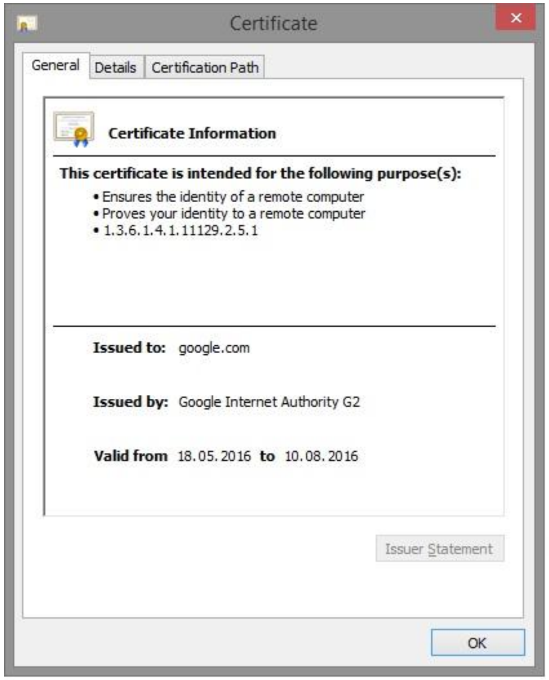
Security strategies



dl.acm.org/doi/pdf/10.1145/3372297.3417252



GlobalSign Root CA - R1
GlobalSign
GlobalSign RSA OV SSL CA 2018
*.bilibili.com



Certificate Information

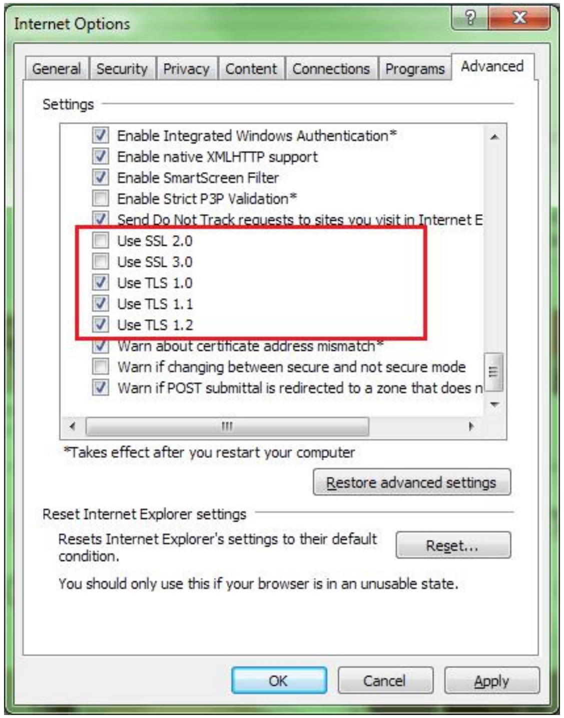
This certificate is intended for the following purpose(s):

- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- 1.3.6.1.4.1.11129.2.5.1

Issued to: google.com

Issued by: Google Internet Authority G2

Valid from 18.05.2016 **to** 10.08.2016



Settings

☑ Enable Integrated Windows Authentication*
☑ Enable native XMLHTTP support
☑ Enable SmartScreen Filter
☐ Enable Strict P3P Validation*
☑ Send Do Not Track requests to sites you visit in Internet E
☐ Use SSL 2.0
☐ Use SSL 3.0
☑ Use TLS 1.0
☑ Use TLS 1.1
☑ Use TLS 1.2
☑ Warn about certificate address mismatch*
☐ Warn if changing between secure and not secure mode
☑ Warn if POST submittal is redirected to a zone that does n

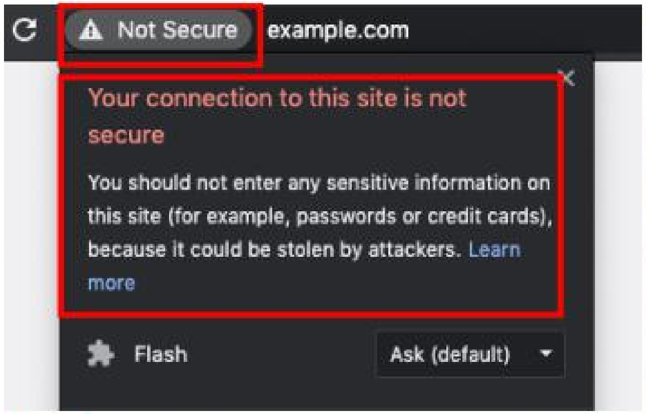
*Takes effect after you restart your computer

Reset Internet Explorer settings

Resets Internet Explorer's settings to their default condition.

You should only use this if your browser is in an unusable state.

Security strategies continued

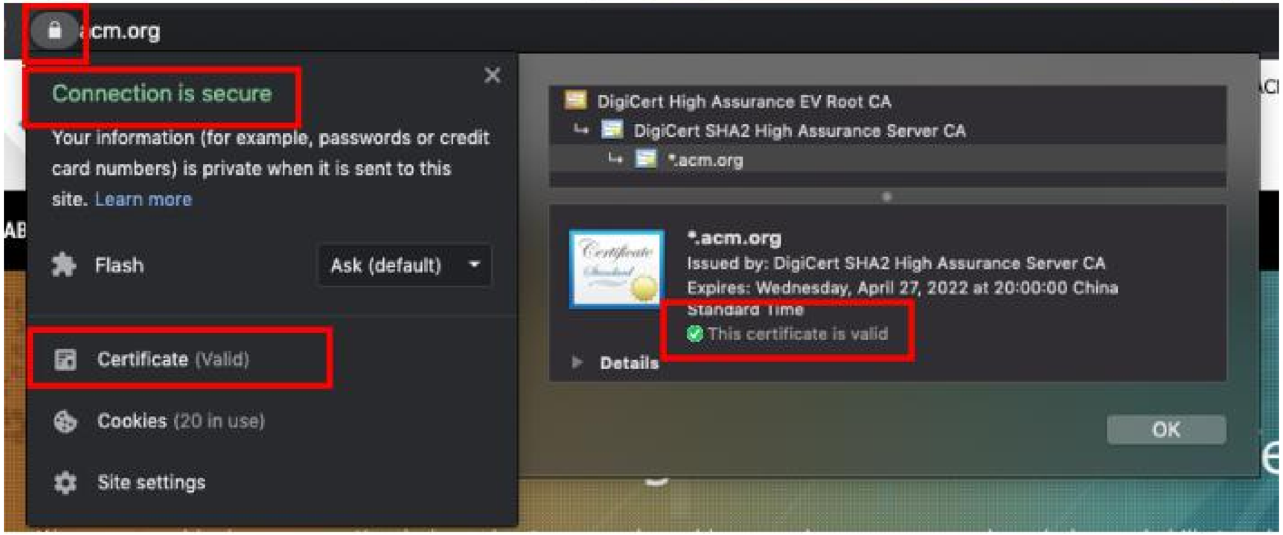


Not Secure example.com

Your connection to this site is not secure

You should not enter any sensitive information on this site (for example, passwords or credit cards), because it could be stolen by attackers. Learn more

Flash Ask (default)



acm.org

Connection is secure

Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

Flash Ask (default)

Certificate (Valid)

Cookies (20 in use)

Site settings

DigiCert High Assurance EV Root CA
↳ DigiCert SHA2 High Assurance Server CA
↳ *.acm.org

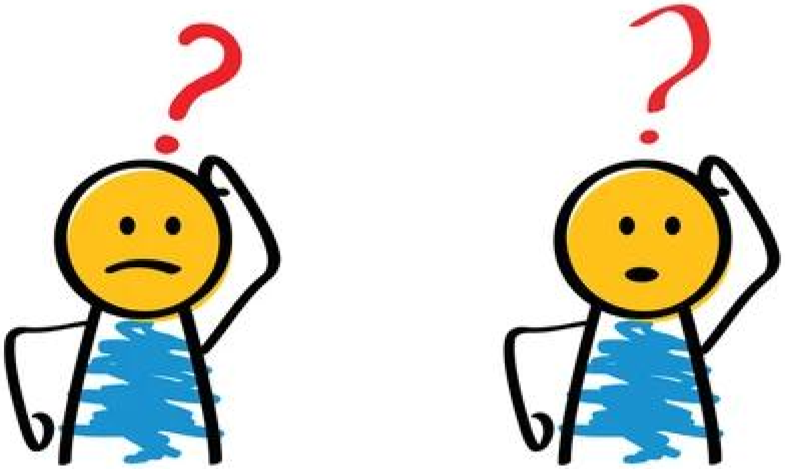
*.acm.org
Issued by: DigiCert SHA2 High Assurance Server CA
Expires: Wednesday, April 27, 2022 at 20:00:00 China Standard Time
✓ This certificate is valid

Details

OK

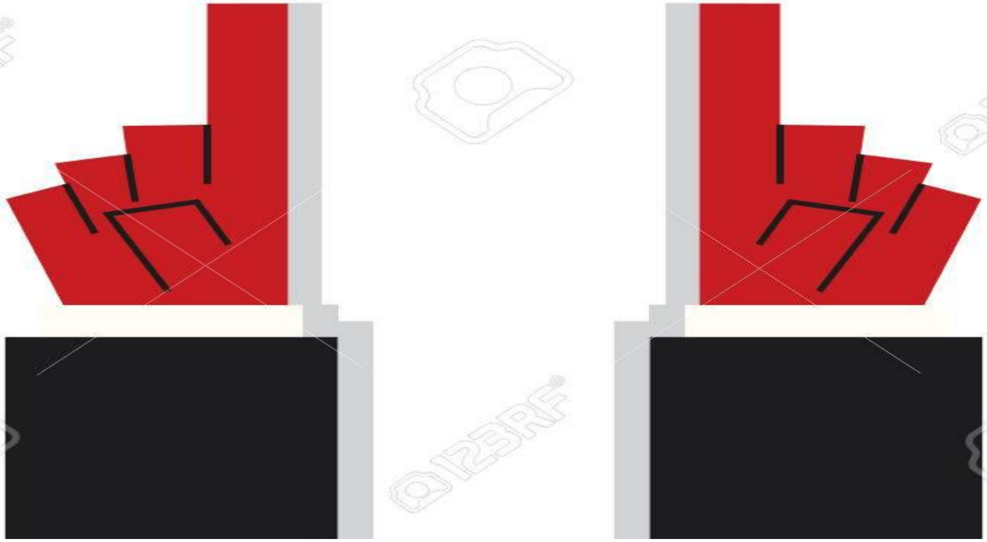
Image from Dr. Mingming Zhang presentation slides, link: https://shenkaiwen.com/files/papers/SCC_CCS20_pre.pdf

But are we really secure enough with these HTTPS-protected websites?



shutterstock.com · 1731904642

OF COURSE NOT

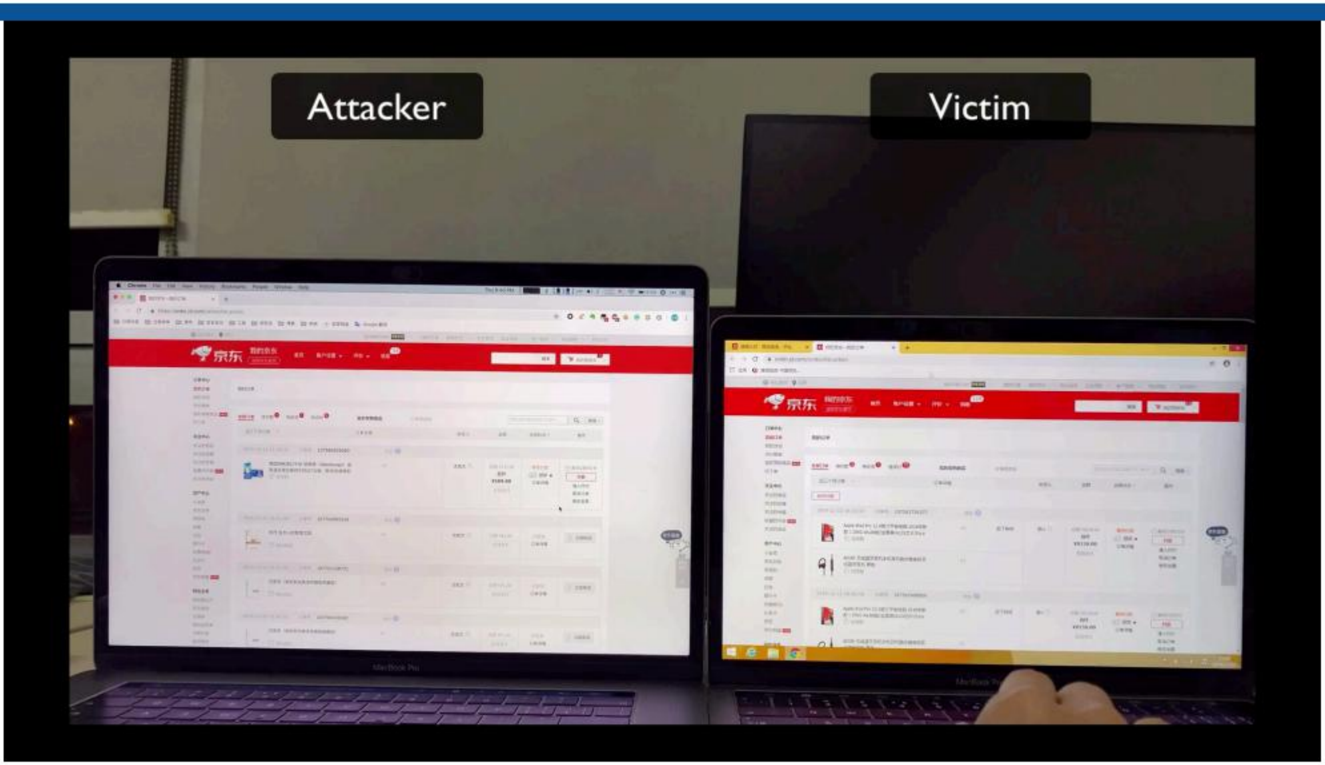


HTTPS Context Confusion Attack (SCC Attack)

Major Components of SCC attack

1. A victim who tries to get access or surf on a webpage
2. A MITM adversary who is able to intercept the network traffic between user and the server
3. A server A with strict security policies
4. A server B with flawed security policies
5. A and B share the same SSL/TLS certificate

Demo: Payment Hijacking on JD Shopping



JD Shopping is a large online shopping website in China (Alexa Rank 10)

Image from Dr. Mingming Zhang presentation slides, link: https://shenkaiwen.com/files/papers/SCC_CCS20_pre.pdf

What SCC attack achives

Unnoticeable by users and also

image from www.dreamstime.com

Undetectable by browers

images from Realworld Graphics

What is SCC attack based on?

- Multiple domains can share certificates.

- These domains that share certificates might have different security policies.

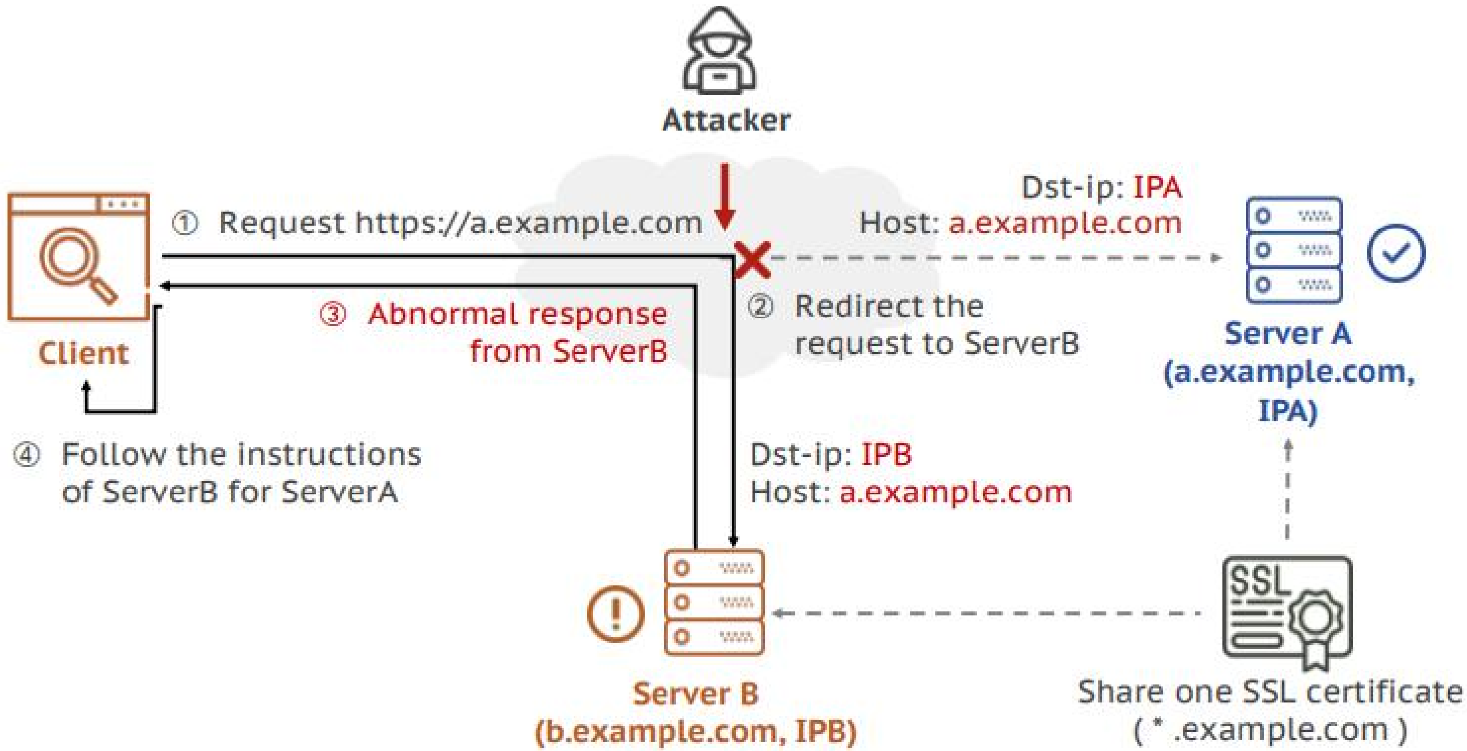
- However the browser only uses the certificate to authenticate the server's identity.

- The attacker can distribute these weak policies to the secure servers and therefore can fool with the security policies of these strong servers.

What harm SCC attacks can cause?

- Internet phishing
- Payment hijacking
- Privacy leakage
- and more!

SCC attack workflow



Attacker
① Request https://a.example.com
Dst-ip: IPA
Host: a.example.com
Client
③ Abnormal response from ServerB
② Redirect the request to ServerB
Server A (a.example.com, IPA)
④ Follow the instructions of ServerB for ServerA
Dst-ip: IPB
Host: a.example.com
Server B (b.example.com, IPB)
SSL
Share one SSL certificate (* .example.com)

Image from Dr. Mingming Zhang presentation slides, link: https://shenkaiwen.com/files/papers/SCC_CCS20_pre.pdf

Types of SCC Attack

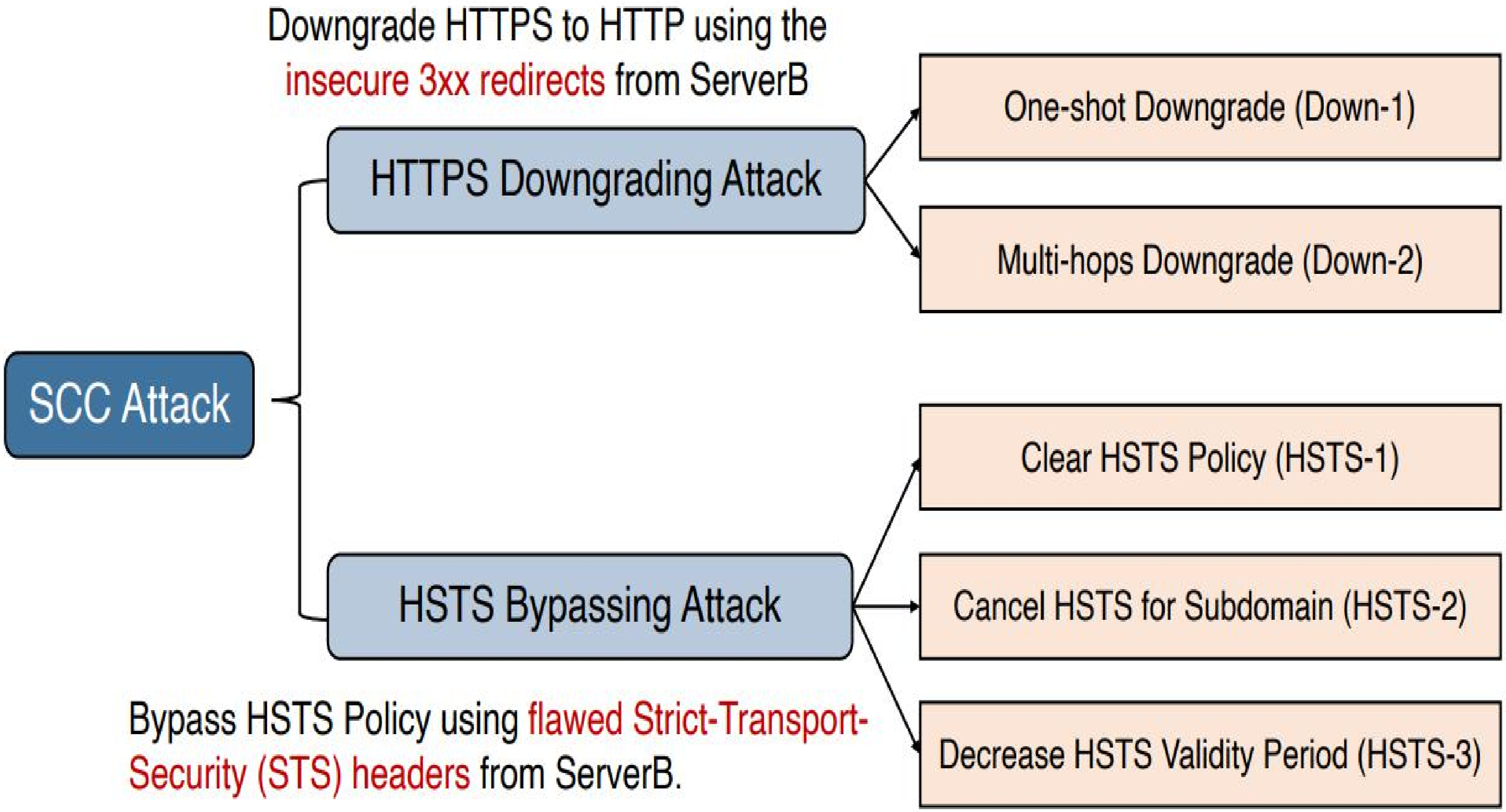

Downgrade HTTPS to HTTP using the insecure 3xx redirects from ServerB
SCC Attack
HTTPS Downgrading Attack
One-shot Downgrade (Down-1)
Multi-hops Downgrade (Down-2)
HSTS Bypassing Attack
Clear HSTS Policy (HSTS-1)
Cancel HSTS for Subdomain (HSTS-2)
Decrease HSTS Validity Period (HSTS-3)
Bypass HSTS Policy using flawed Strict-Transport-Security (STS) headers from ServerB.

Image from Dr. Mingming Zhang presentation slides, link: https://shenkaiwen.com/files/papers/SCC_CCS20_pre.pdf

Downgrading attacks

One-shot downgrading attack

Reroute the network traffic to a server with flawed policies which shares the same certificate, if the flawed server returns a 3xx redirect to some HTTP url, the attack is done.

Multi-hop downgrading attack

In cases where no server sharing the same certificate returns insecure 3xx redirects, the attacker can perform a range of reroutes the server that returns insecure redirects.

Three types of bypasses

1. set HSTS <max-age = 0>

- If the max-age setting is 0 then the server will clear all HSTS policies for serverA

2. set HSTS <no includeSubdomain>

- If the setting is <no includeSubdomain> then the server will not apply the same HSTS policies in its subdomains therefore the protection of the subdomains will be cancelled.

3. Max-age= <smaller-than-Server A>

- If the max-age of server B is set smaller than A then the validity period of A will likewise be decreased.

Main causes of SCC attack

- Certifcate sharing
- Levels of security of certificate sharing servers are decided by the most flawed server, the so called “bucket effects”.

Bucket Effect

image from Porta-Fi

Are there any solutions?

Unfortunately

Certificate sharing is a very mature technology and is now used for almosts all websites. Due to considerations from various aspects, it is unlikely that we can cease the use of certificate sharing.

Possible mitigations

Server-side

- Enhance the security policies of flawed servers
- Unitize the security strategies for websites that share certificates

Browser-side

- Notify the user when any insecure context change happens at browser-side
- Block all HTTP requests sent from HTTPs websites

Criticizm

- The articled analyzed what could have been done at browser-side and server-side for mitigation.

- However, the authors of the article did not carefully consider what preventions could have been done at client-side.

- In fact, internet browsers such as Google Chrome and Firefox are not the only tools users can rely on.

Encrypt at user-side

- Users can use end-to-end encryptions to prevent their plaintext input from being "sniffed"
- Such as using Diffie-Hellman key exchange, RSA, AES, etc.
- Thus, the data will be encrypted before transmission.
- What the attacker can "sniff" are now ciphertexts instead of plaintext.

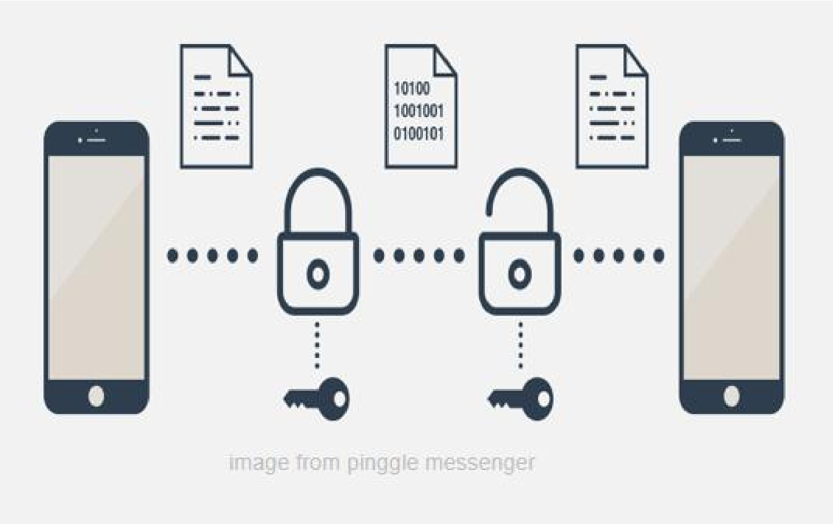


image from pinggle messenger

Setting to Skip HTTP redirects

Initial connection problem:

The browser must establish an HTTPS connection with the server at least once to enable HSTS, If the middleman keeps preventing the browser from establishing the initial HTTPS connection with the server, HSTS will not be able to function.

Solution:

Embed the HSTS site list into the browser.

Solves the initial connection problem:

As long as the browser determines that HSTS is enabled on the site, it will skip the original HTTP redirection and directly initiate an HTTPS request.

Mitigates the insecure redirection problem:

This way, the browser will automatically activates HSTS protection and all HTTP redirects and requests will be blocked.

Link to the original paper:

https://dl.acm.org/doi/pdf/10.1145/3372297.3417252

Link to Dr.Mingming Zhang’s presentation slides for the paper:

https://shenkaiwen.com/files/papers/SCC_CCS20_pre.pdf

Link to Dr Mingming Zhang’s presentation recording

https://www.bilibili.com/video/BV1F54y1678H?from=search&seid=1195356197771623273&spm_id_from=333.337.0.0

Thank you!