Shifting Knowledge Left

Keeping Up With Modern Application Security

Mark Stanislav

Head of Security Engineering

Fletcher Heisler

CEO / Founder

HUNTER2

Overview

- The State of Developer Security Knowledge
- The Need to Reduce Time-to-Education
- A Thoughtful Approach to Engineer Enablement
- Changing Course on Education
- Growing the Community

The State of Developer Security Knowledge

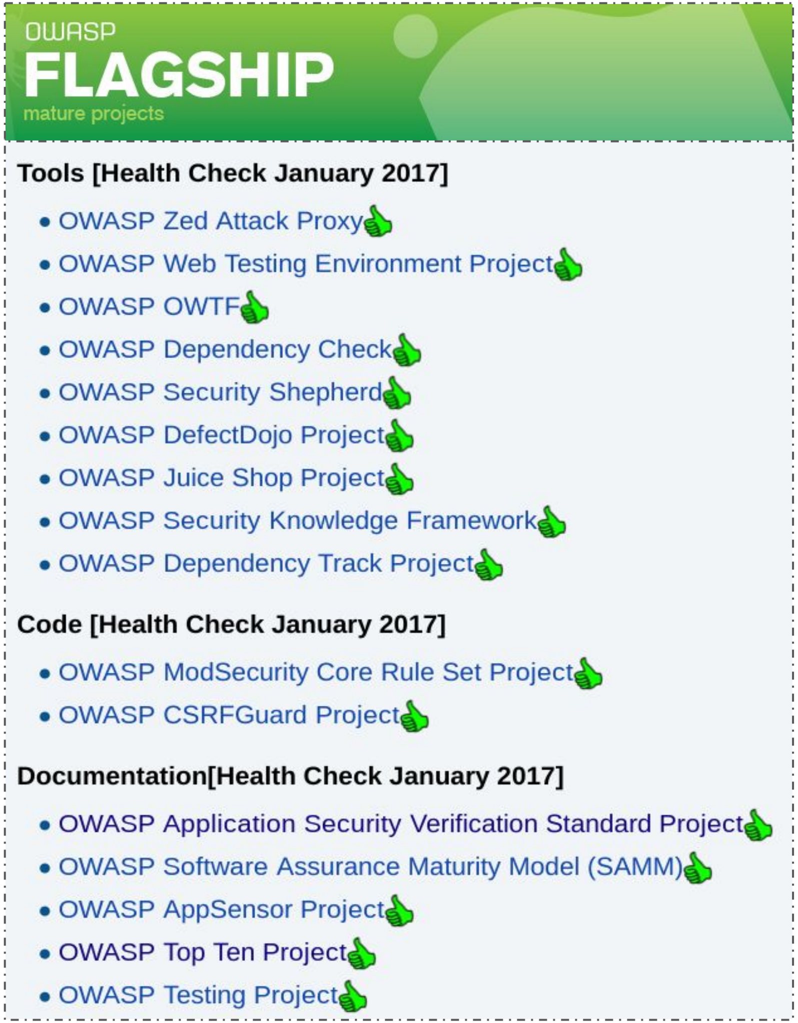
“The **OWASP Top 10** is a powerful awareness document for web application security. It represents a broad consensus about the most critical security risks to web applications.”

- OWASP

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Over 125 OWASP Projects...

- 60% Are Currently “active”
- 13% Are Flagship Projects



OWASP

FLAGSHIP

mature projects

Tools [Health Check January 2017]

- OWASP Zed Attack Proxy👍
- OWASP Web Testing Environment Project👍
- OWASP OWTF👍
- OWASP Dependency Check👍
- OWASP Security Shepherd👍
- OWASP DefectDojo Project👍
- OWASP Juice Shop Project👍
- OWASP Security Knowledge Framework👍
- OWASP Dependency Track Project👍

Code [Health Check January 2017]

- OWASP ModSecurity Core Rule Set Project👍
- OWASP CSRFGuard Project👍

Documentation[Health Check January 2017]

- OWASP Application Security Verification Standard Project👍
- OWASP Software Assurance Maturity Model (SAMM)👍
- OWASP AppSensor Project👍
- OWASP Top Ten Project👍
- OWASP Testing Project👍

> “Nearly one in five developers are not at all familiar with the Top 10 OWASP application security risks.”
>
> - Veracode

https://techbeacon.com/security/32-application-security-stats-matter

The OWASP Top 10 is Not…

- Up to date
- Language- or framework-specific
- A checklist for code scanning and pentesting
- An exhaustive list of vulnerability classes
- A training syllabus

Top U.S. Computer Science Programs

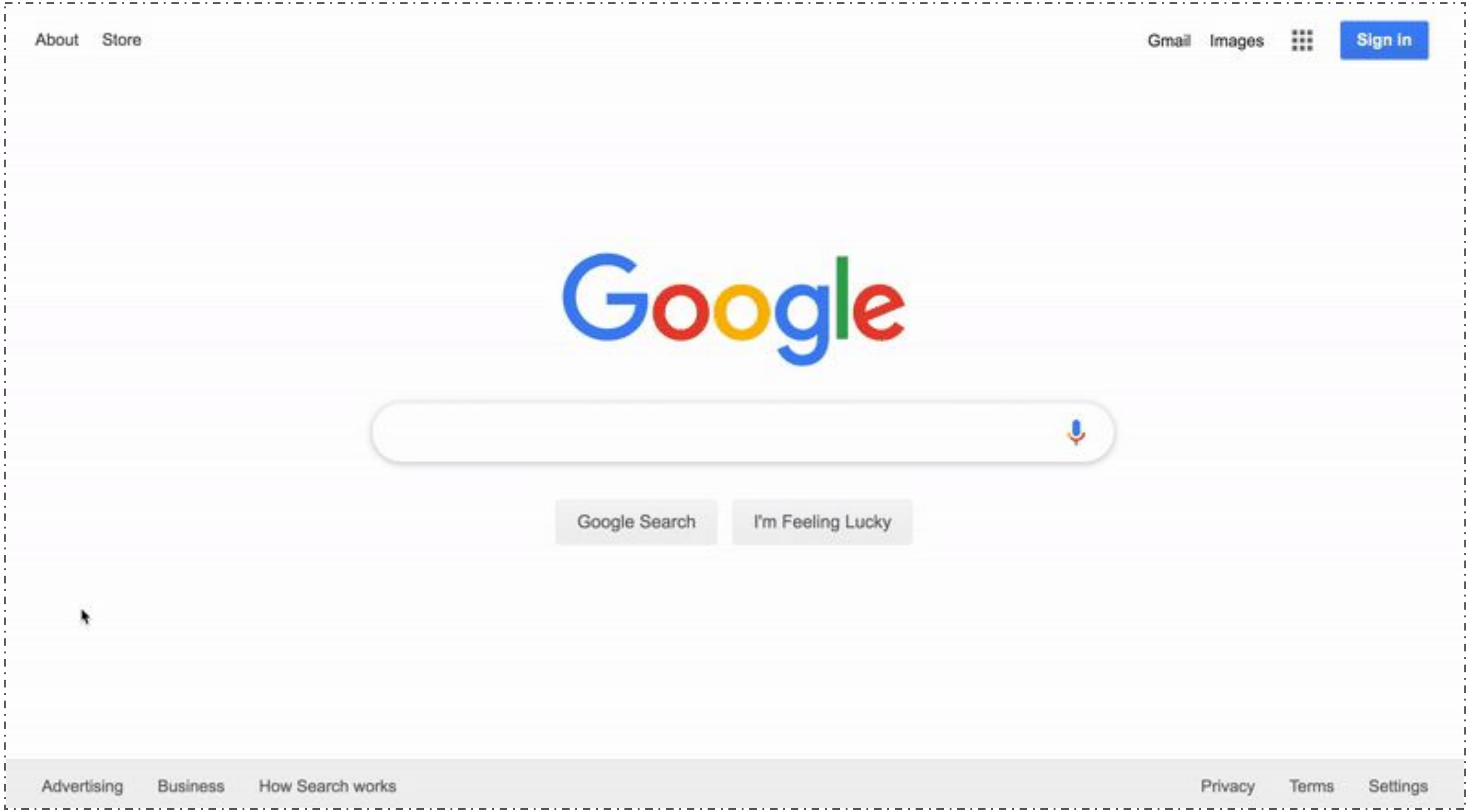
1. Carnegie Mellon
2. MIT
3. Stanford
4. University of California, Berkeley
5. University of Illinois, Urbana-Champaign
6. Cornell
7. University of Washington
8. Georgia Tech
9. Princeton
10. University of Texas at Austin

https://www.usnews.com/best-graduate-schools/top-science-schools/computer-science-rankings

Top U.S. Computer Science Programs Requiring a Course Related to Software Security:

[This slide left intentionally blank.]

A Moment in the Life of a Developer...



DevSecOps: Doing More With Less!

Industry trends continue to ask engineers to take on more areas of responsibility:

70% of developers are “expected” to write secure code, but…

< 50% of these developers receive feedback on security, and…

25% think their organization's security practices are "good."

https://www.darkreading.com/application-security/software-developers-face-secure-coding-challenges/d/d-id/1335247

https://about.gitlab.com/2019/07/15/global-developer-report/

Dumbing Down Topics = Expanding Risk

Typical Developer Training:

- “Just Use These headers”
- “Just Use the ORM”
- “Just Use This Package”
- Static, Out-of-date Content
- Infrequent (e.g. Annual)

Real Code Security:

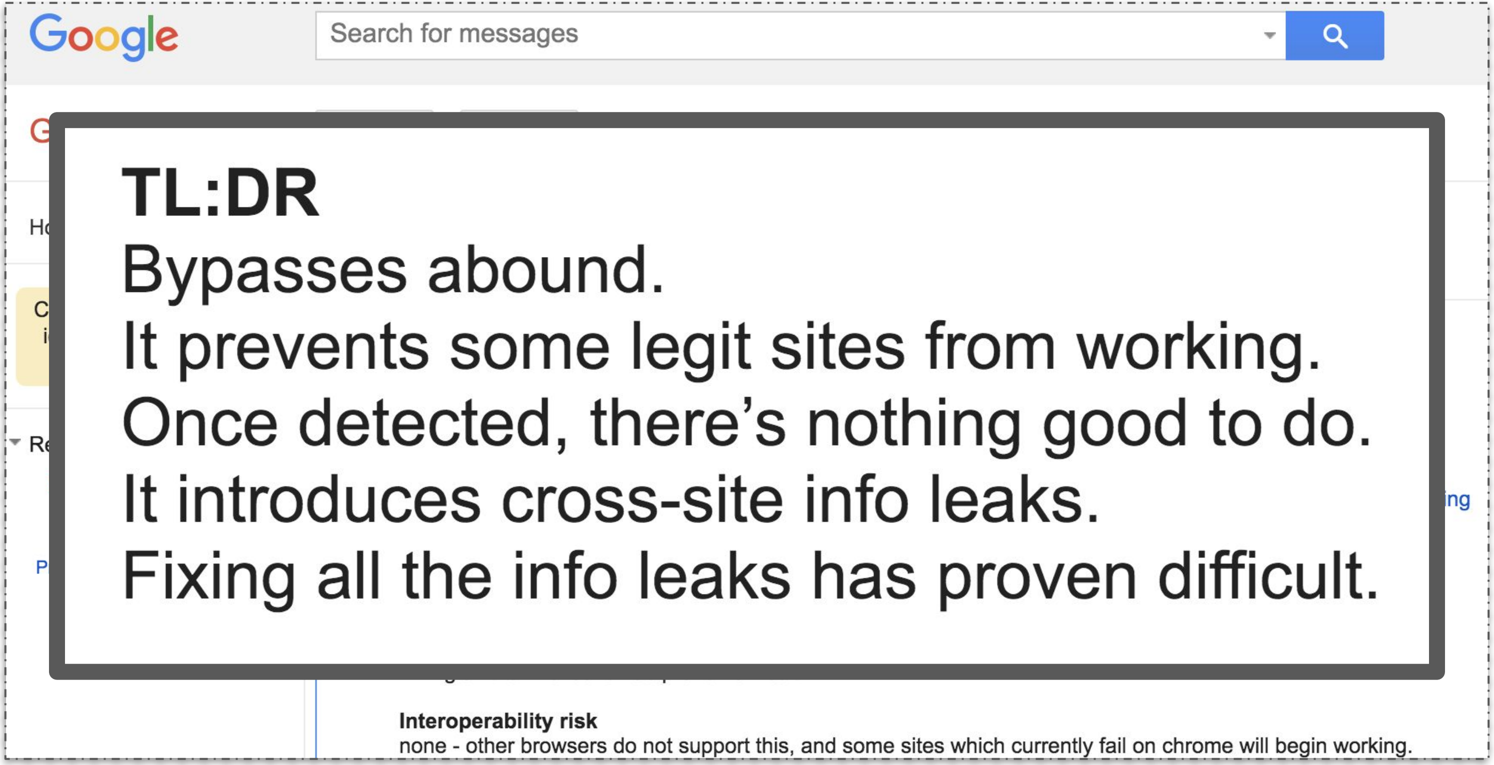
- Defense-in-Depth
- Modern Controls
- Practical Trade-offs
- Threat Modeling
- “Best Practices” Evolve

Load a Metasploit Module → I Can Pentest

=

Use This Browser Header → I Can Prevent XSS

In Browsers We Trust: XSSAuditor



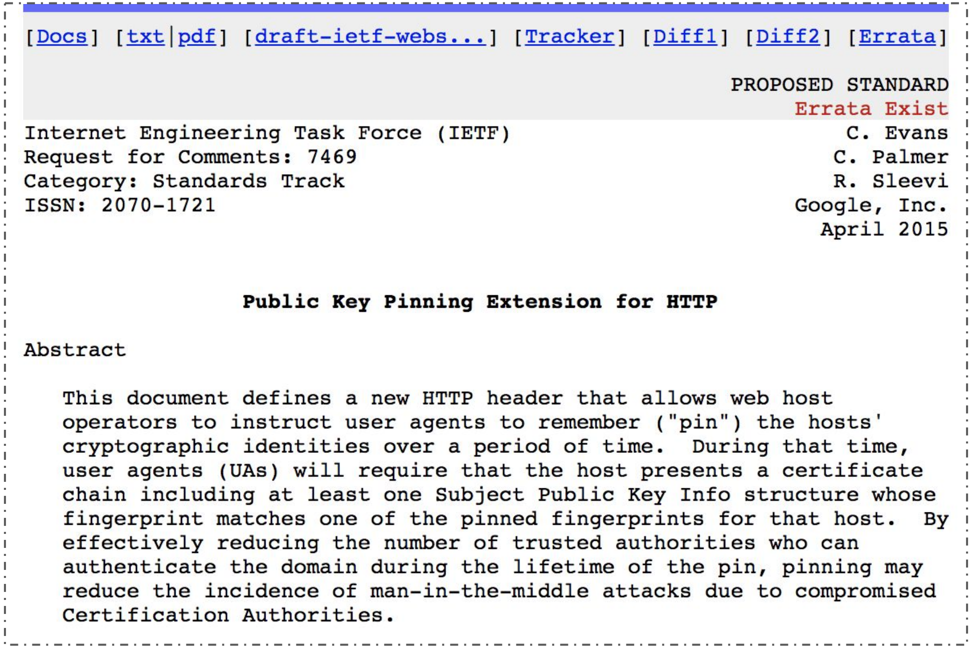
TL:DR
Bypasses abound.
It prevents some legit sites from working.
Once detected, there’s nothing good to do.
It introduces cross-site info leaks.
Fixing all the info leaks has proven difficult.

Interoperability risk
none - other browsers do not support this, and some sites which currently fail on chrome will begin working.

https://groups.google.com/a/chromium.org/forum/#!msg/blink-dev/TuYw-EZhO9g/blGViehIAwAJ

HPKP Timeline

04/2015: RFC



[Docs] [txt|pdf] [draft-ietf-webs...] [Tracker] [Diff1] [Diff2] [Errata]

```
                                                       PROPOSED STANDARD
                                                            Errata Exist
Internet Engineering Task Force (IETF)                          C. Evans
Request for Comments: 7469                                     C. Palmer
Category: Standards Track                                       R. Sleevi
ISSN: 2070-1721                                              Google, Inc.
                                                              April 2015
```

Public Key Pinning Extension for HTTP

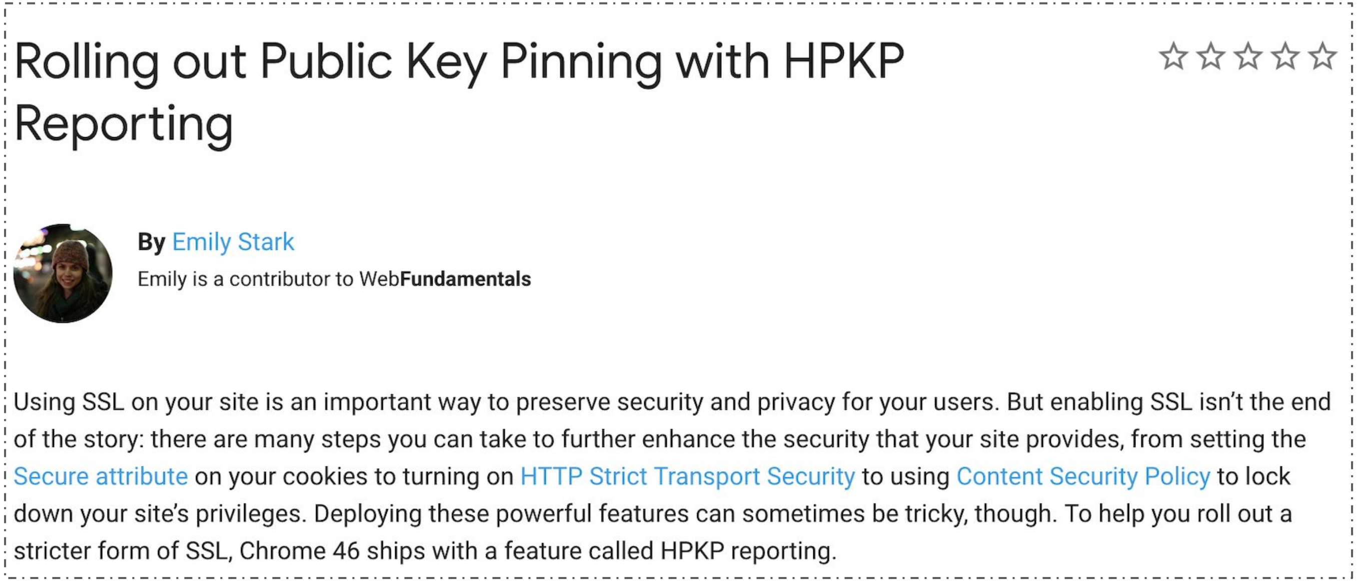
Abstract

This document defines a new HTTP header that allows web host operators to instruct user agents to remember ("pin") the hosts' cryptographic identities over a period of time. During that time, user agents (UAs) will require that the host presents a certificate chain including at least one Subject Public Key Info structure whose fingerprint matches one of the pinned fingerprints for that host. By effectively reducing the number of trusted authorities who can authenticate the domain during the lifetime of the pin, pinning may reduce the incidence of man-in-the-middle attacks due to compromised Certification Authorities.

https://tools.ietf.org/html/rfc7469

HPKP Timeline, cont.

09/2015: Chrome rollout



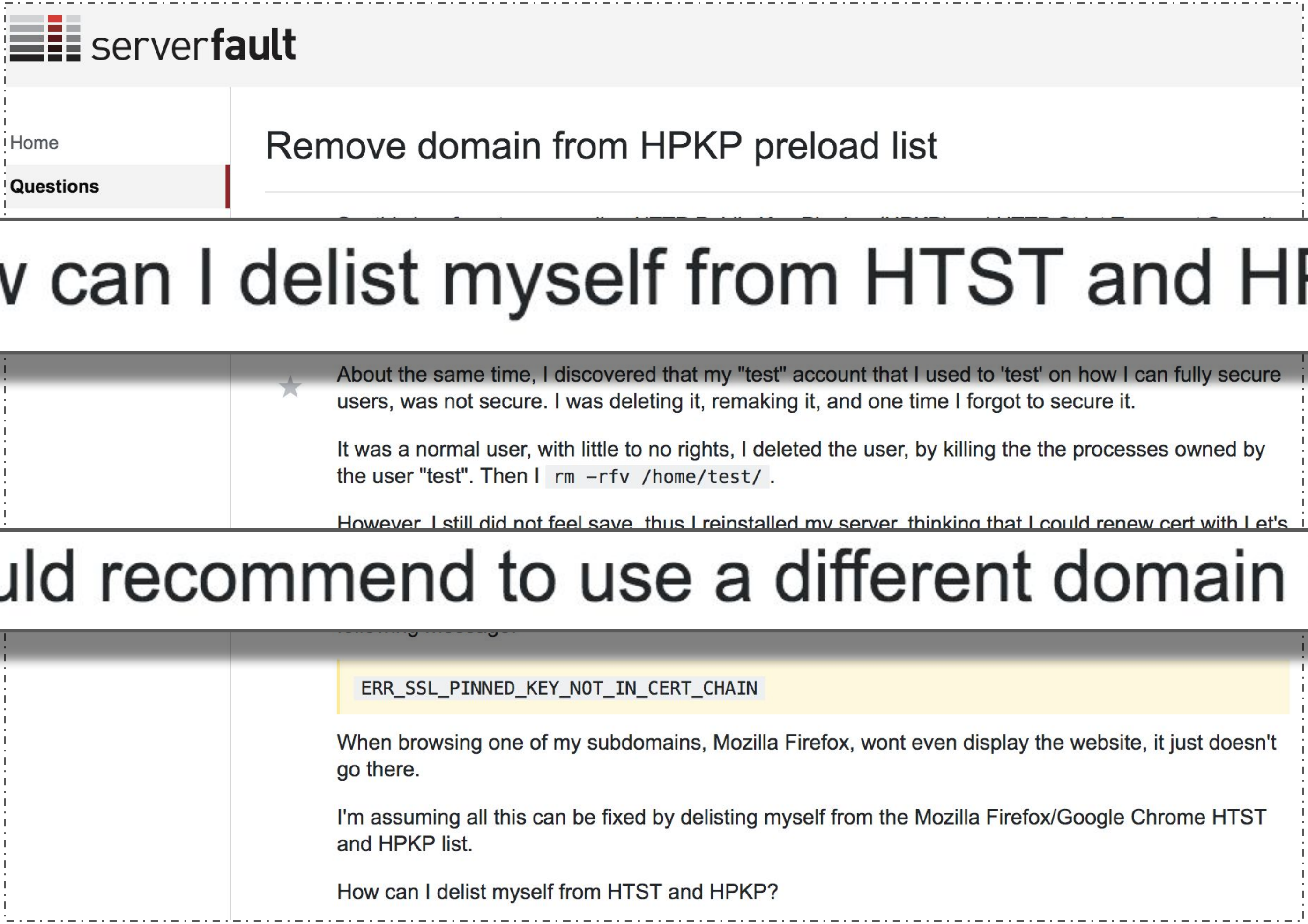
Rolling out Public Key Pinning with HPKP Reporting

By Emily Stark

Emily is a contributor to Web**Fundamentals**

Using SSL on your site is an important way to preserve security and privacy for your users. But enabling SSL isn't the end of the story: there are many steps you can take to further enhance the security that your site provides, from setting the Secure attribute on your cookies to turning on HTTP Strict Transport Security to using Content Security Policy to lock down your site's privileges. Deploying these powerful features can sometimes be tricky, though. To help you roll out a stricter form of SSL, Chrome 46 ships with a feature called HPKP reporting.

https://developers.google.com/web/updates/2015/09/HPKP-reporting-with-chrome-46?hl=bg



Remove domain from HPKP preload list

How can I delist myself from HTST and HPKP?

About the same time, I discovered that my "test" account that I used to 'test' on how I can fully secure users, was not secure. I was deleting it, remaking it, and one time I forgot to secure it.

It was a normal user, with little to no rights, I deleted the user, by killing the the processes owned by the user "test". Then I `rm -rfv /home/test/`.

However, I still did not feel save, thus I reinstalled my server, thinking that I could renew cert with Let's

I would recommend to use a different domain name.

```
ERR_SSL_PINNED_KEY_NOT_IN_CERT_CHAIN
```

When browsing one of my subdomains, Mozilla Firefox, wont even display the website, it just doesn't go there.

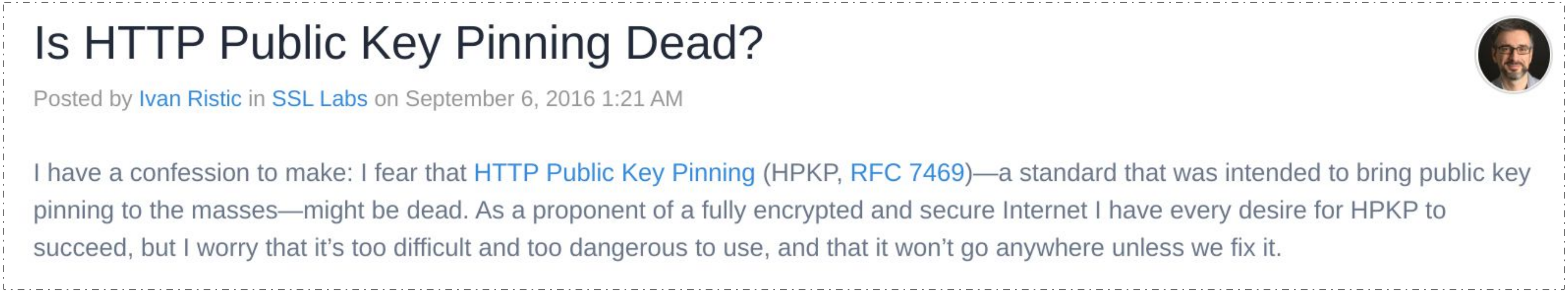
I'm assuming all this can be fixed by delisting myself from the Mozilla Firefox/Google Chrome HTST and HPKP list.

How can I delist myself from HTST and HPKP?

https://serverfault.com/questions/835797/remove-domain-from-hpkp-preload-list

HPKP Timeline, cont.

09/2016:



Is HTTP Public Key Pinning Dead?

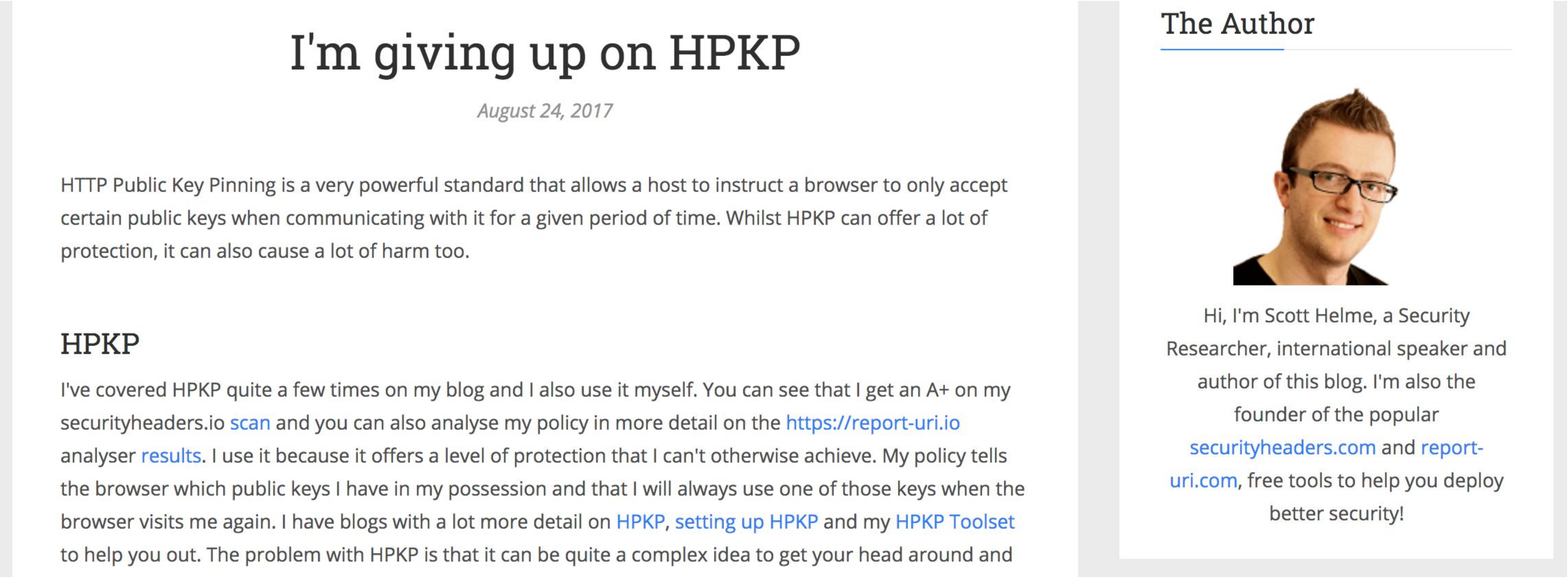
Posted by Ivan Ristic in SSL Labs on September 6, 2016 1:21 AM

I have a confession to make: I fear that HTTP Public Key Pinning (HPKP, RFC 7469)—a standard that was intended to bring public key pinning to the masses—might be dead. As a proponent of a fully encrypted and secure Internet I have every desire for HPKP to succeed, but I worry that it's too difficult and too dangerous to use, and that it won't go anywhere unless we fix it.

https://blog.qualys.com/ssllabs/2016/09/06/is-http-public-key-pinning-dead

HPKP Timeline, cont.

08/2017:



I'm giving up on HPKP

August 24, 2017

HTTP Public Key Pinning is a very powerful standard that allows a host to instruct a browser to only accept certain public keys when communicating with it for a given period of time. Whilst HPKP can offer a lot of protection, it can also cause a lot of harm too.

HPKP

I've covered HPKP quite a few times on my blog and I also use it myself. You can see that I get an A+ on my securityheaders.io scan and you can also analyse my policy in more detail on the https://report-uri.io analyser results. I use it because it offers a level of protection that I can't otherwise achieve. My policy tells the browser which public keys I have in my possession and that I will always use one of those keys when the browser visits me again. I have blogs with a lot more detail on HPKP, setting up HPKP and my HPKP Toolset to help you out. The problem with HPKP is that it can be quite a complex idea to get your head around and

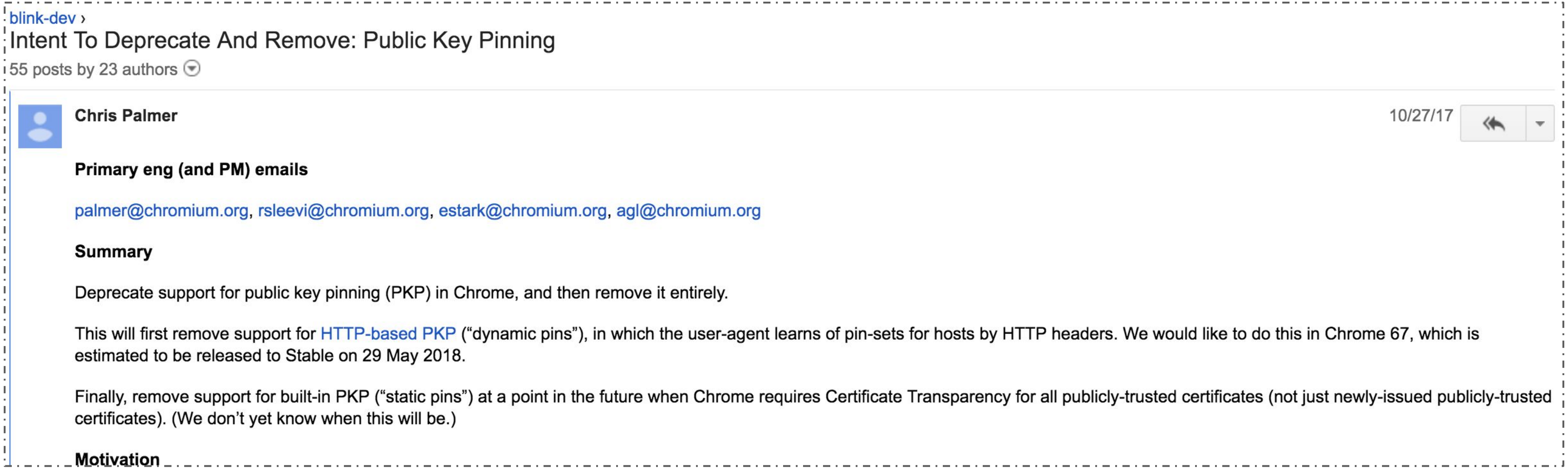
The Author

Hi, I'm Scott Helme, a Security Researcher, international speaker and author of this blog. I'm also the founder of the popular securityheaders.com and report-uri.com, free tools to help you deploy better security!

https://scotthelme.co.uk/im-giving-up-on-hpkp/

HPKP Timeline, cont.

10/2017: Intent to deprecate



Intent To Deprecate And Remove: Public Key Pinning

55 posts by 23 authors

Chris Palmer 10/27/17

Primary eng (and PM) emails

palmer@chromium.org, rsleevi@chromium.org, estark@chromium.org, agl@chromium.org

Summary

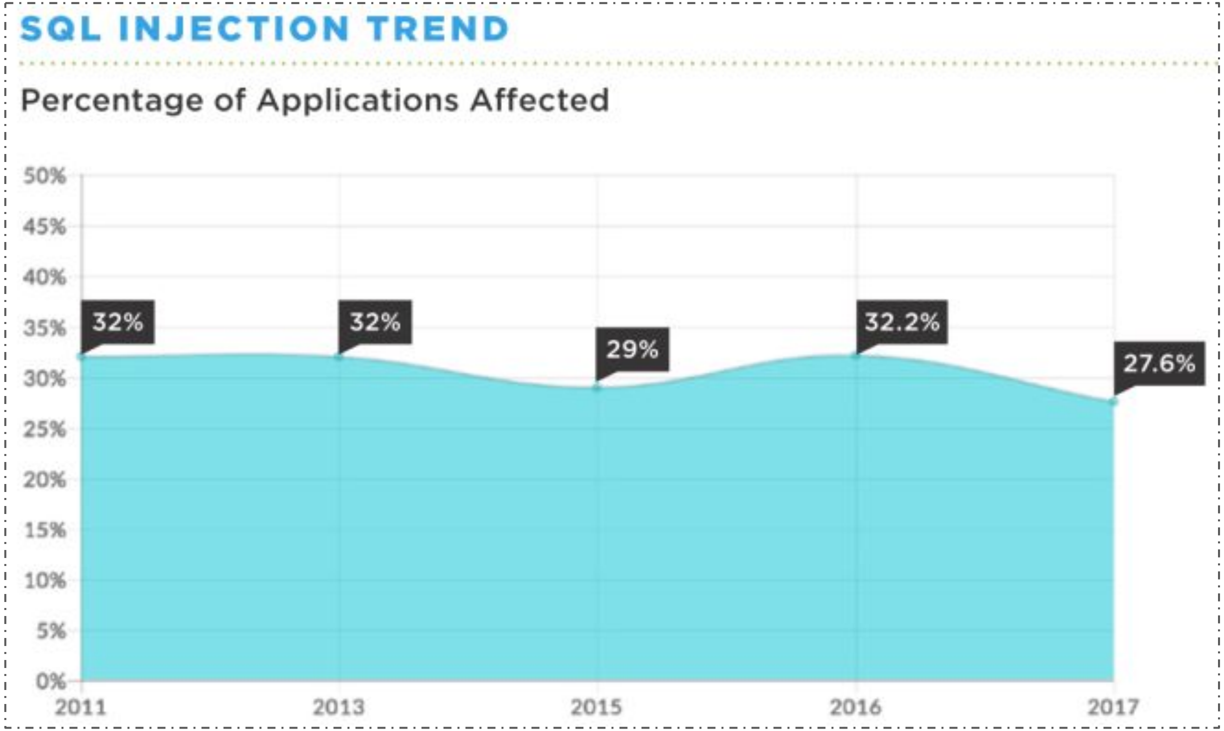
Deprecate support for public key pinning (PKP) in Chrome, and then remove it entirely.

This will first remove support for HTTP-based PKP ("dynamic pins"), in which the user-agent learns of pin-sets for hosts by HTTP headers. We would like to do this in Chrome 67, which is estimated to be released to Stable on 29 May 2018.

Finally, remove support for built-in PKP ("static pins") at a point in the future when Chrome requires Certificate Transparency for all publicly-trusted certificates (not just newly-issued publicly-trusted certificates). (We don't yet know when this will be.)

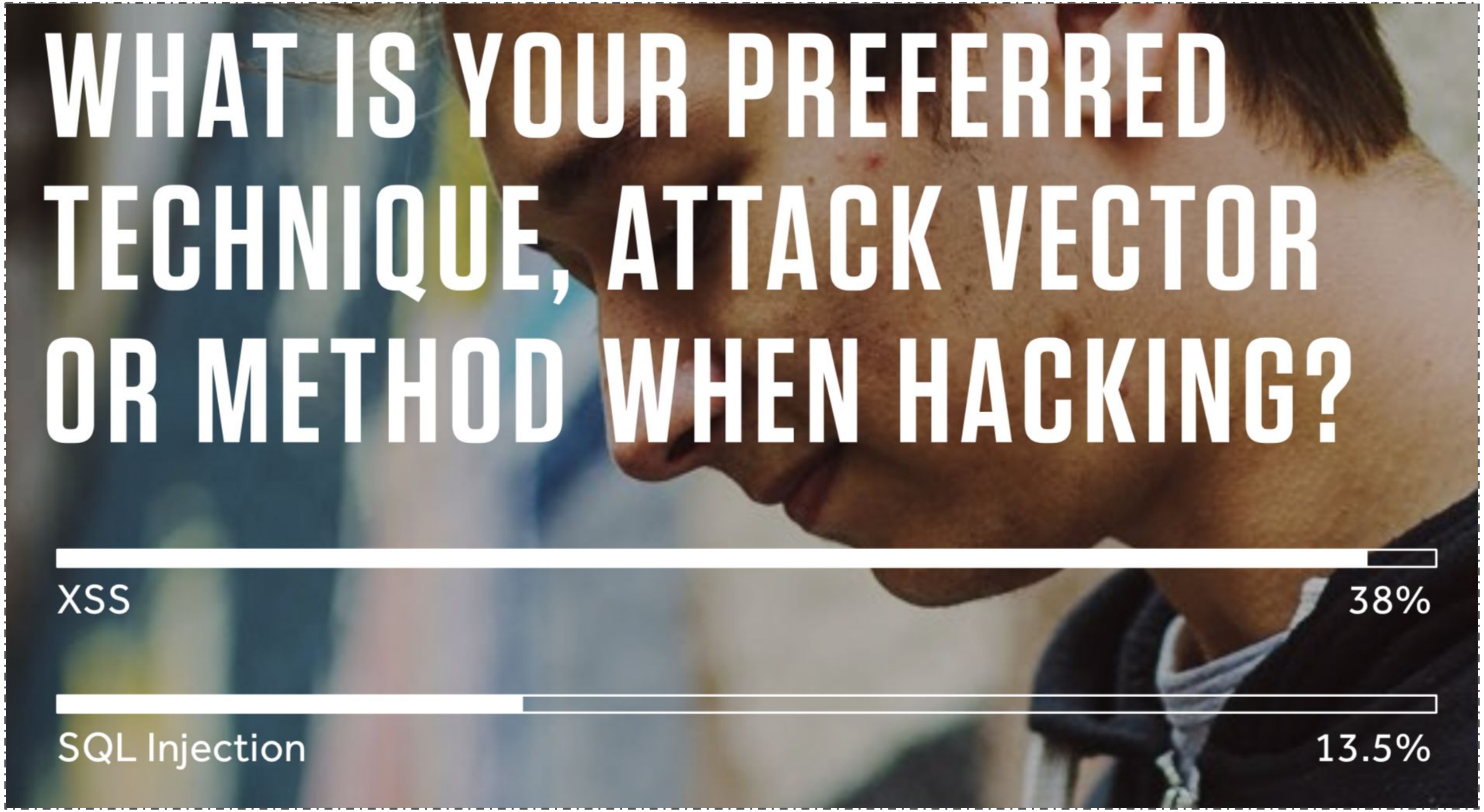
Motivation

https://groups.google.com/a/chromium.org/forum/#!msg/blink-dev/he9tr7p3rZ8/eNMwKPmUBAAJ



“The pass rate of applications against standards like the OWASP Top 10 hasn’t budged in recent years, with applications failing policy consistently around 70% of the time.” - Veracode

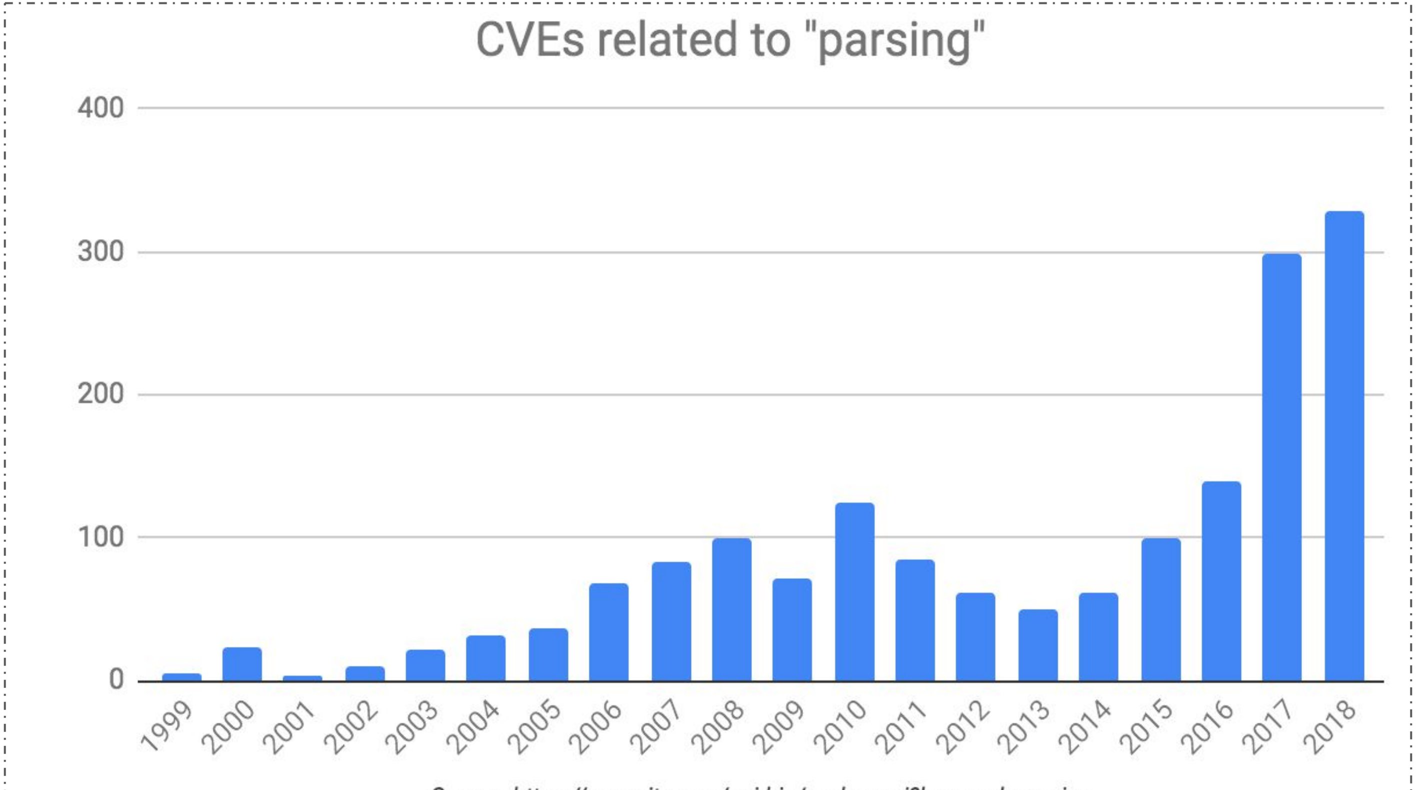
https://www.veracode.com/blog/secure-development/what-developers-need-know-about-state-software-security-today



“XSS continues to be the most common weakness type no matter how it’s measured.” - HackerOne

https://www.hackerone.com/resources/top-10-vulnerabilities

More Code, More Problems



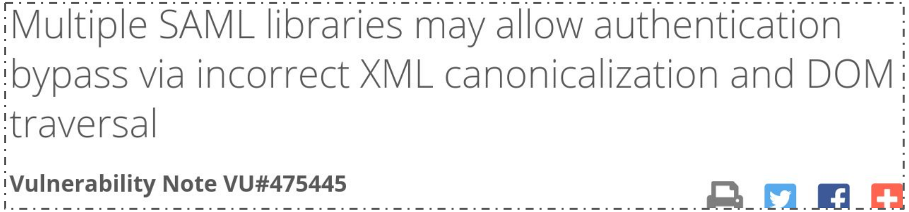
Source: https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=parsing

Wishful Thinking as Vulnerability Management

Identity Theft

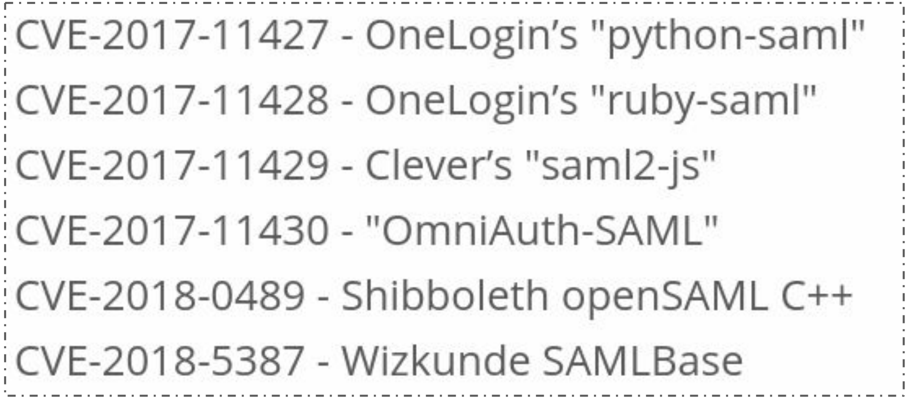
Attacks on Modern SSO Systems

DUO LABS



Multiple SAML libraries may allow authentication bypass via incorrect XML canonicalization and DOM traversal

Vulnerability Note VU#475445



CVE-2017-11427 - OneLogin's "python-saml"

CVE-2017-11428 - OneLogin's "ruby-saml"

CVE-2017-11429 - Clever's "saml2-js"

CVE-2017-11430 - "OmniAuth-SAML"

CVE-2018-0489 - Shibboleth openSAML C++

CVE-2018-5387 - Wizkunde SAMLBase

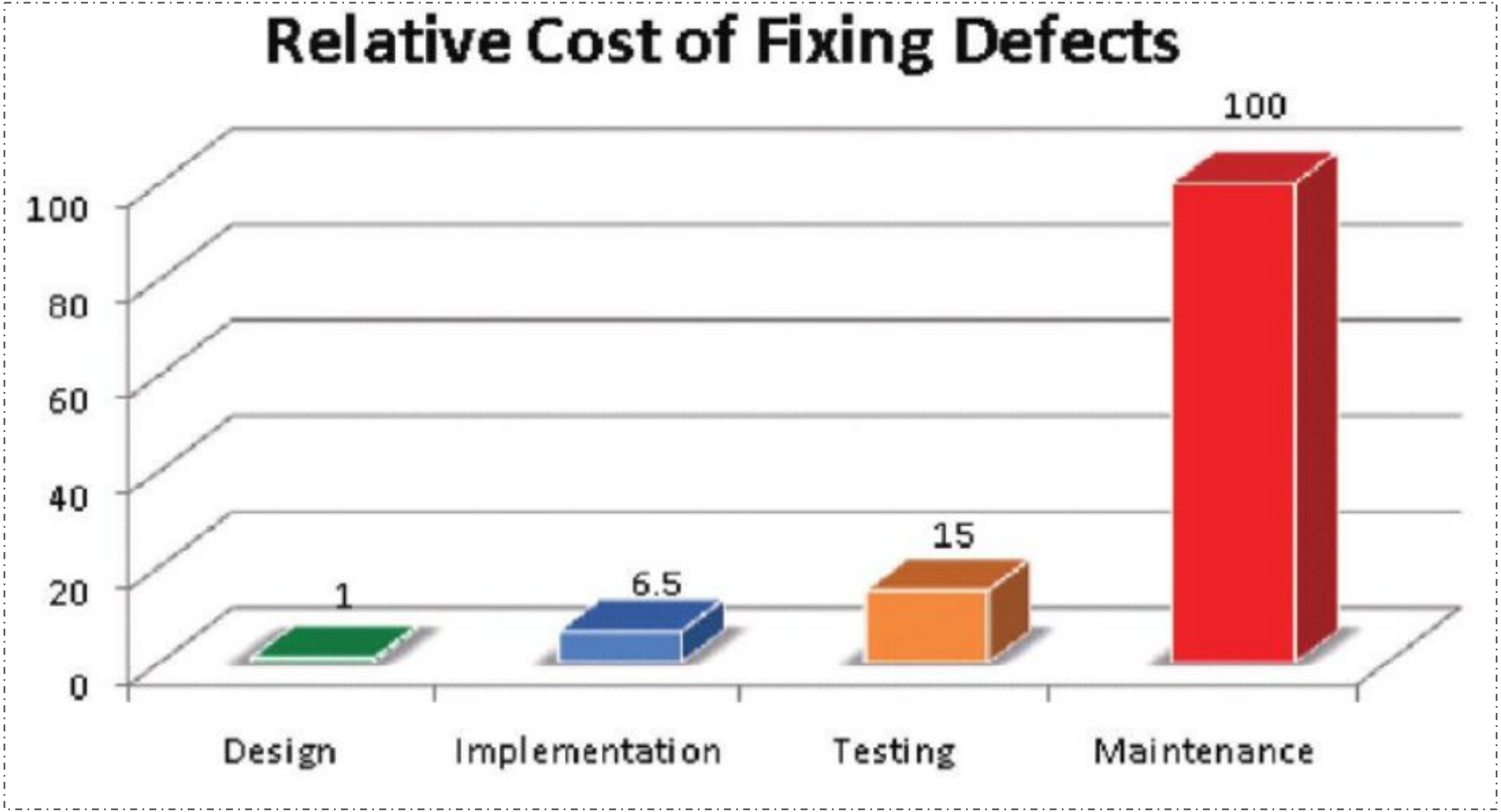
"We aren't vulnerable because we don't use those libraries..."

“You can’t scan your way to secure code.”

- P. Pourmousa, Veracode

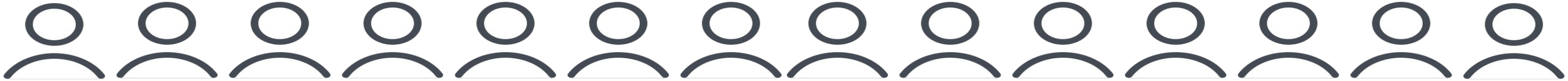
https://www.veracode.com/blog/managing-appsec/beyond-scanning-dont-let-appsec-ignorance-become-negligence

The Need to Reduce Time-to-Education

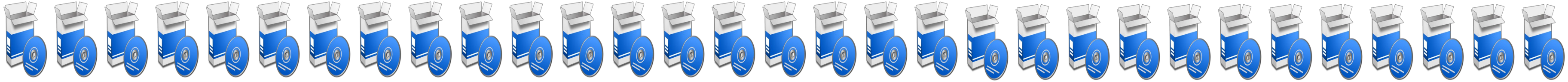


https://www.researchgate.net/publication/255965523_Integrating_Software_Assurance_into_the_Software_Development_Life_Cycle_SDLC

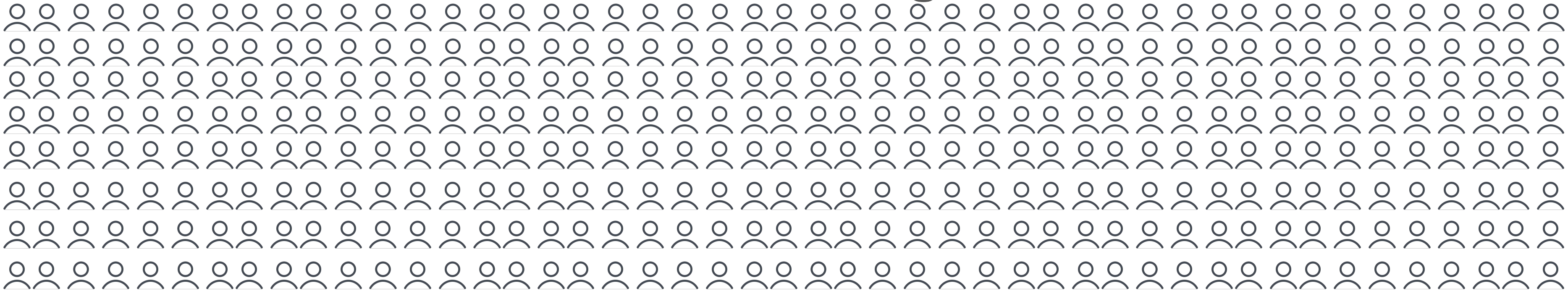
Industry Compliance



Products

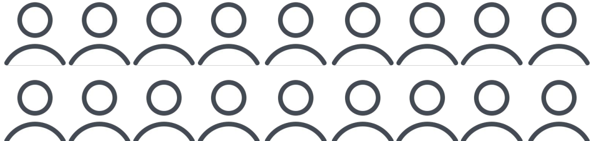


Software Engineers



Security Engineers

Pentesters



SAST Triage

Risk Versus Reward

“Vulnerabilities that fall into the SSRF IDOR categories earn some of the higher bounties given the risk they pose to an organization.”
- HackerOne

Duo New Engineer Survey

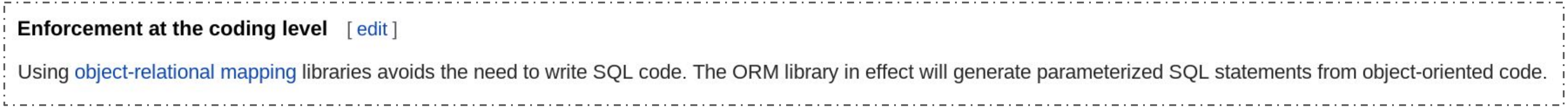
How familiar are you with the following vulnerability classes?

SSRF: 58% **not** familiar at all
IDOR: 67% **not** familiar at all

“There is 40% crossover of the HackerOne Top 10 to the latest version of the OWASP Top 10.” - HackerOne

https://www.hackerone.com/resources/top-10-vulnerabilities

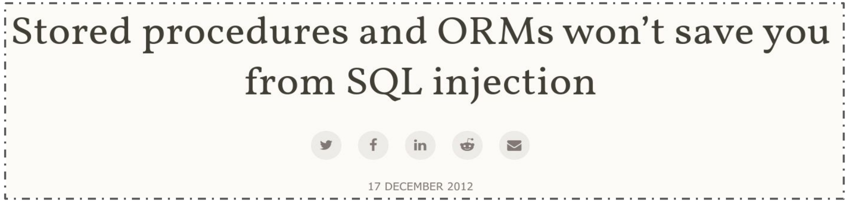
ORM: Not SQLi Proof!



Enforcement at the coding level [edit]

Using object-relational mapping libraries avoids the need to write SQL code. The ORM library in effect will generate parameterized SQL statements from object-oriented code.

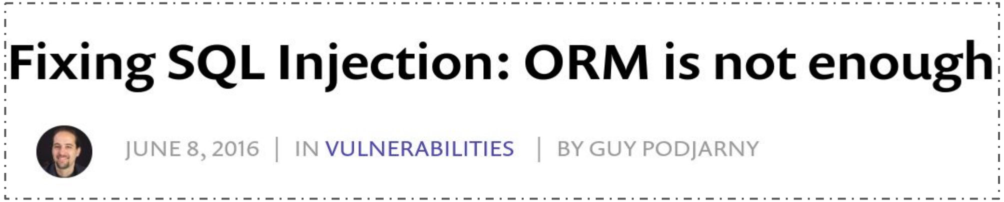
https://en.wikipedia.org/wiki/SQL_injection#Mitigation



Stored procedures and ORMs won't save you from SQL injection

17 DECEMBER 2012

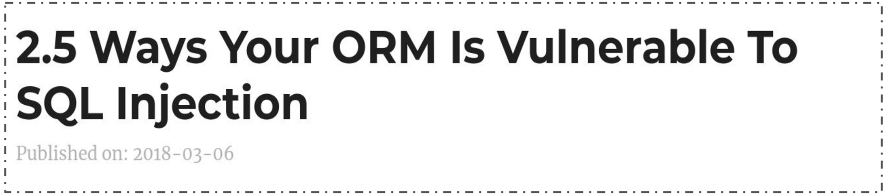
https://www.troyhunt.com/stored-procedures-and-orms-wont-save/



Fixing SQL Injection: ORM is not enough

JUNE 8, 2016 | IN VULNERABILITIES | BY GUY PODJARNY

https://snyk.io/blog/sql-injection-orm-vulnerabilities/

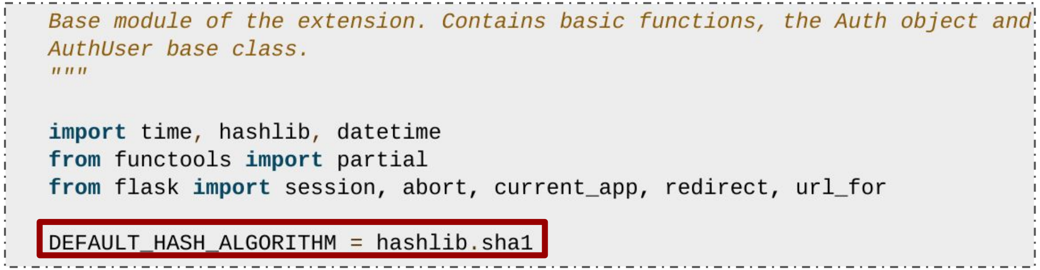


2.5 Ways Your ORM Is Vulnerable To SQL Injection

Published on: 2018-03-06

https://bertwagner.com/2018/03/06/2-5-ways-your-orm-will-allow-sql-injection/

Education at the Speed of Reality?

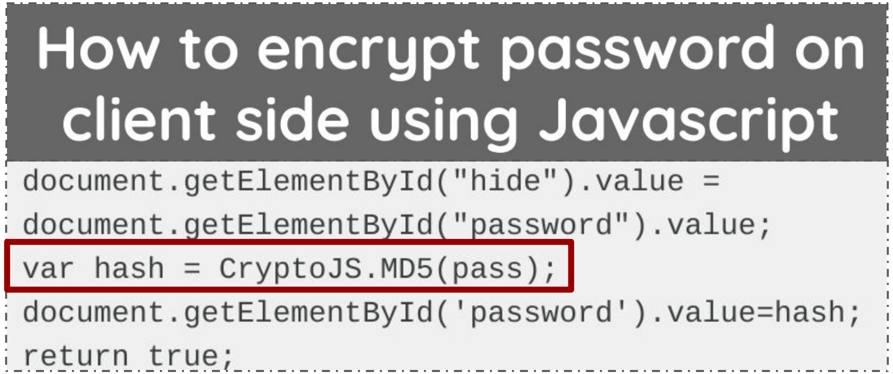


```
Base module of the extension. Contains basic functions, the Auth object and
AuthUser base class.
"""

import time, hashlib, datetime
from functools import partial
from flask import session, abort, current_app, redirect, url_for

DEFAULT_HASH_ALGORITHM = hashlib.sha1
```

2011

https://pythonhosted.org/Flask-Auth/_modules/flaskext/auth/auth.html

2019

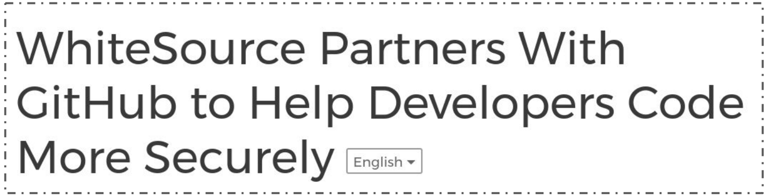


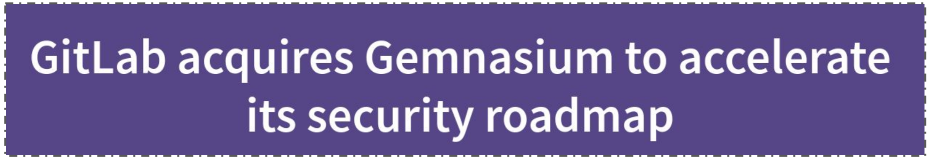
How to encrypt password on client side using Javascript

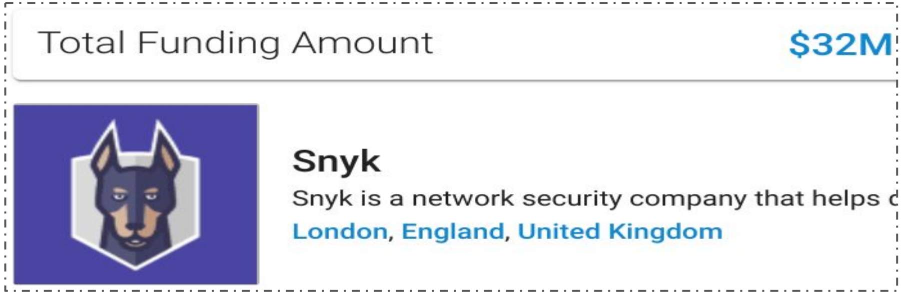
```
document.getElementById("hide").value =
document.getElementById("password").value;
var hash = CryptoJS.MD5(pass);
document.getElementById('password').value=hash;
return true;
```

bcrypt:	1999
PBKDF2:	2000
scrypt:	2009
Argon2:	2015

If a Vulnerability Gets Flagged… Now What?


WhiteSource Partners With GitHub to Help Developers Code More Securely

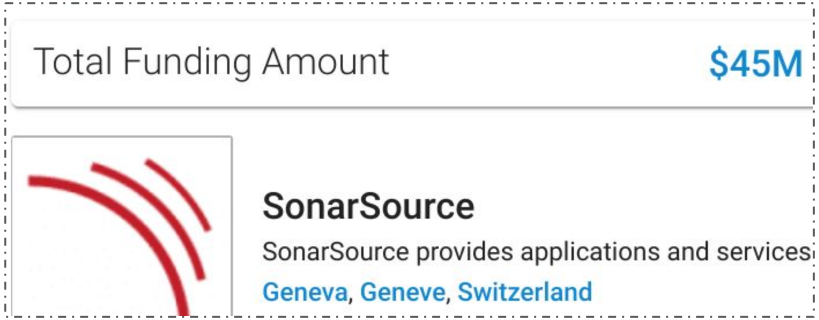

GitLab acquires Gemnasium to accelerate its security roadmap


Total Funding Amount $32M

Snyk

Snyk is a network security company that helps c

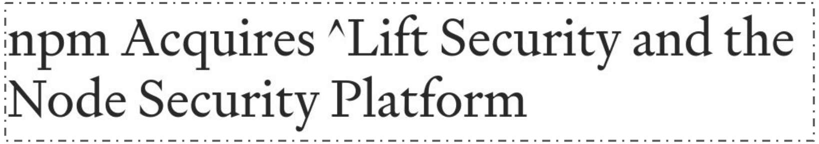
London, England, United Kingdom

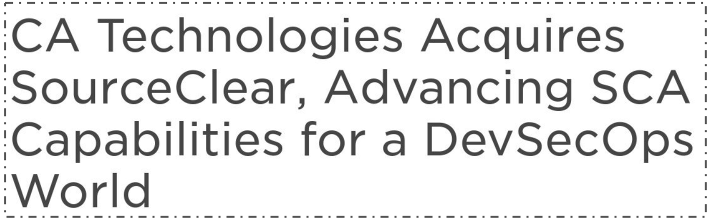

Total Funding Amount $45M

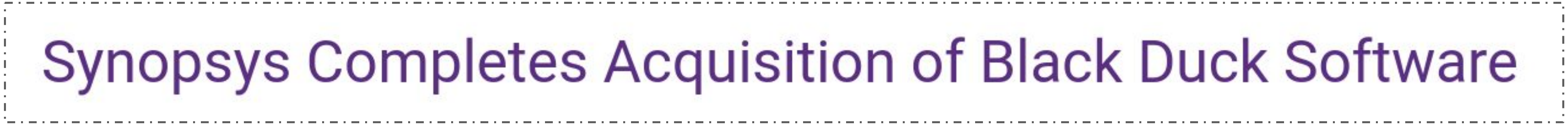
SonarSource

SonarSource provides applications and services

Geneva, Geneve, Switzerland


npm Acquires ^Lift Security and the Node Security Platform


CA Technologies Acquires SourceClear, Advancing SCA Capabilities for a DevSecOps World


Synopsys Completes Acquisition of Black Duck Software

A Thoughtful Approach to Engineer Enablement

OH: Security Conference Talk

Engineers may say that you punish them for bugs found; so we should ask them ‘Why aren’t you good at coding?’

Meanwhile, the presenter is...

- Brand new to application security
- Has never been a software engineer
- Admits to not having any real knowledge of programming

But sure, be an Application Security Engineer ¯_(ツ)_/¯

Centering Team Focus Beyond “Find Bugs”

Engineering is Family → Adversarial in Action, Not Relationship

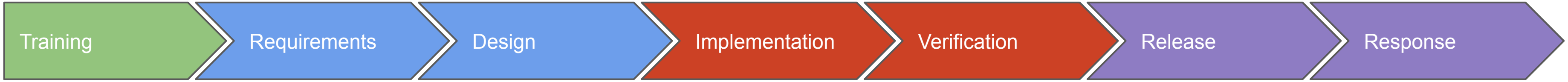
Low Friction, High Value → Elegance to Obviate Engineer Frustration

Build a Paved Road → Spend Time Enabling Good Outcomes

How Could it Go Right? → Meet the Need for Innovation, Not FUD

No Code Left Behind → Take Inventory, Know the Risk, Clean Up

Rethinking the Security Development Lifecycle



Training

Requirements

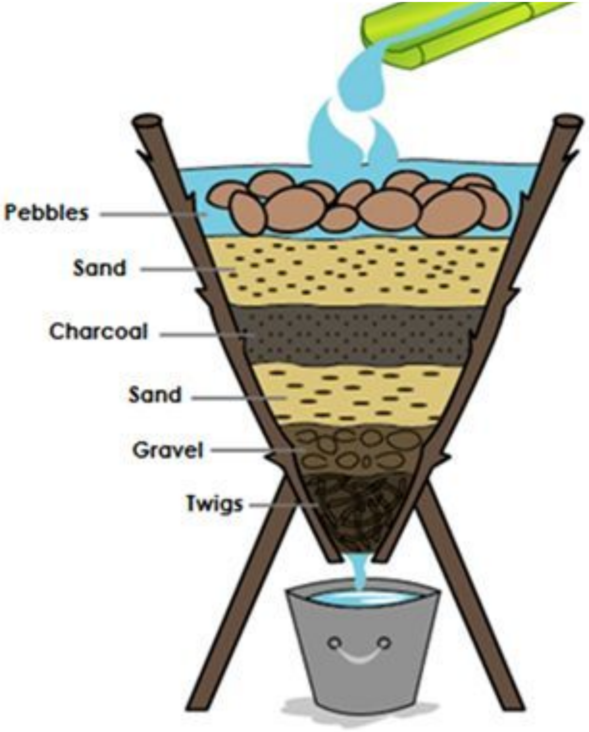
Design

Implementation

Verification

Release

Response



← *This*

Training

Requirements

Design

Implementation

Verification

Release

Response

Not →

Many Front Doors to Enablement

In-person (or WebEx)

Office Hours - Weekly

Visit Team Meetings - Monthly

Training Courses - Quarterly

Internal CTF - Annual

Guest Speakers - Annual

Online/Digital

Hunter2 - Self Service

SDL Guidelines - Self Service

Slack #appsec - On Demand

psirt@duo.com - On Demand

Security Pipeline - On Demand

Raise the Bar for Your Engineers

An “OWASP Top 10” Training Usually Results in…

1. ' OR '1'='1'
2. <script>alert(‘hacked’);</script>
3. ../../../../../etc/passwd

Challenge your engineers by sharing content that is not something they have already seen ad-nauseum!

Introduction(?) to Application Security at Duo

Encrypted Cookies Are Not Enough

- Developers often encrypt cookie payloads assuming it cannot be changed
- Encryption **does not provide integrity**! Attackers can modify an encrypted cookie without knowing the key

```
def encryptCookie(payload, key, iv):
    obj = AES.new(key, AES.MODE_CTR, iv)
    str1 = padding(payload)
    ciphertext = obj.encrypt(str1)
    return ciphertext

AuthCookieVal = encryptCookie("Role:Reviewer", "aiBuacoM8", "mee0epJee")
```

Bit-flip to Victory

Cookie payload = "Role:**Reviewer**" provides the cookie value (hex) below

```
set-cookie: auth=de6dd89e66232da8a4dac92845;
```

This isn't signed!

Attacker:

By gathering cookies from various roles, looking for patterns and bit-flipping with XOR, a new valid cookie can be crafted without knowing the encryption key

```
de6dd89e66232da8a4dac92845 XOR 13011b000b
```

```
Cookie: auth=de6dd89e66302cb3a4d1
```

Outcome used to set attacker's cookie

Decrypts to "Role:**Admin**"

“I had other app security training with the previous jobs and this one is the best so far. The labs make it particularly fun and engaging.”

“It was great! I'd love if there were more beyond the 3 [trainings]!”

3 In-house Built Courses

Each Course Runs Quarterly

141 Attendees Across Classes

No Required Attendance

An AppSec Office Hours Anecdote

Engineer: *“What is the right encryption choice for these LDAP secrets?”*

↓

AppSec Team: *“Hmm… what feature are you working on that requires that?”*

↓

Engineer: *[Interesting new functionality that we were not yet aware of...]*

↓

AppSec Team: *“Gotcha! Let’s take a step back and review the design with you.”*

Meet the Engineers Where *They* Work

Be **Predictable**

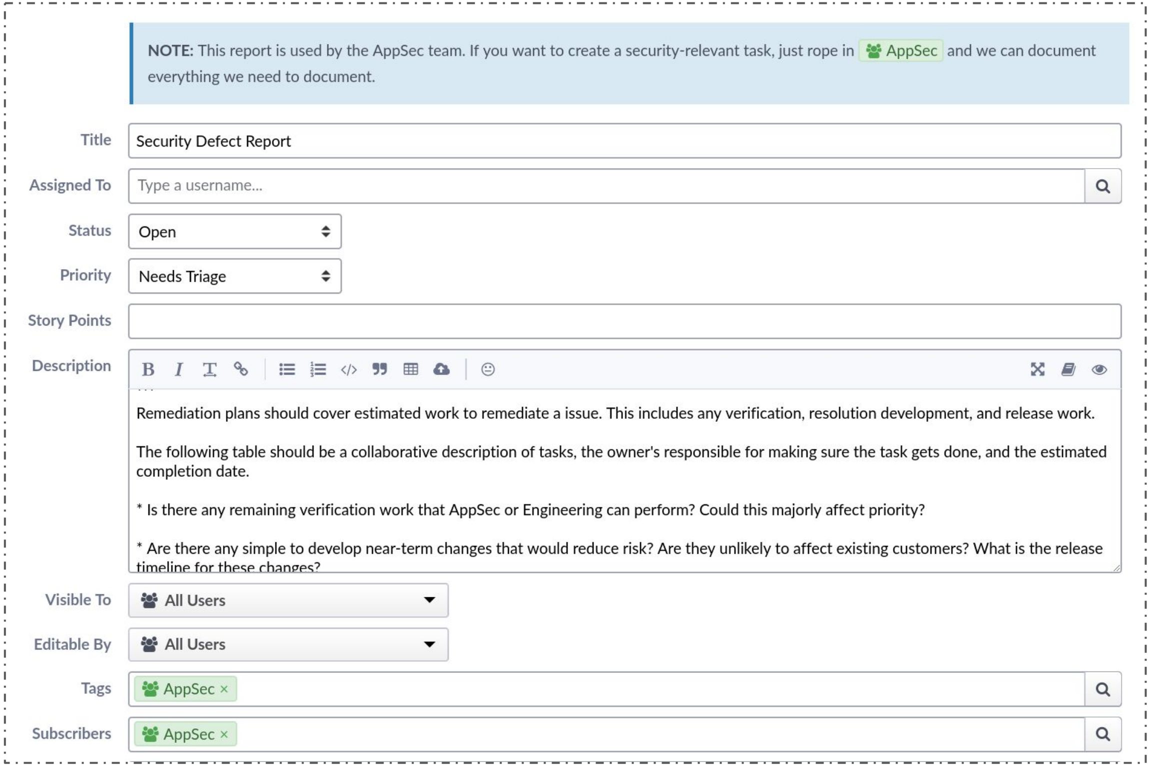
Communicate Well

Share **Context**

Explain **Risk**

Suggest **Remediation**

Support Next Steps



NOTE: This report is used by the AppSec team. If you want to create a security-relevant task, just rope in AppSec and we can document everything we need to document.

Title	Security Defect Report
Assigned To	Type a username...
Status	Open
Priority	Needs Triage
Story Points	
Description	Remediation plans should cover estimated work to remediate a issue. This includes any verification, resolution development, and release work. The following table should be a collaborative description of tasks, the owner's responsible for making sure the task gets done, and the estimated completion date. * Is there any remaining verification work that AppSec or Engineering can perform? Could this majorly affect priority? * Are there any simple to develop near-term changes that would reduce risk? Are they unlikely to affect existing customers? What is the release timeline for these changes?
Visible To	All Users
Editable By	All Users
Tags	AppSec ×
Subscribers	AppSec ×

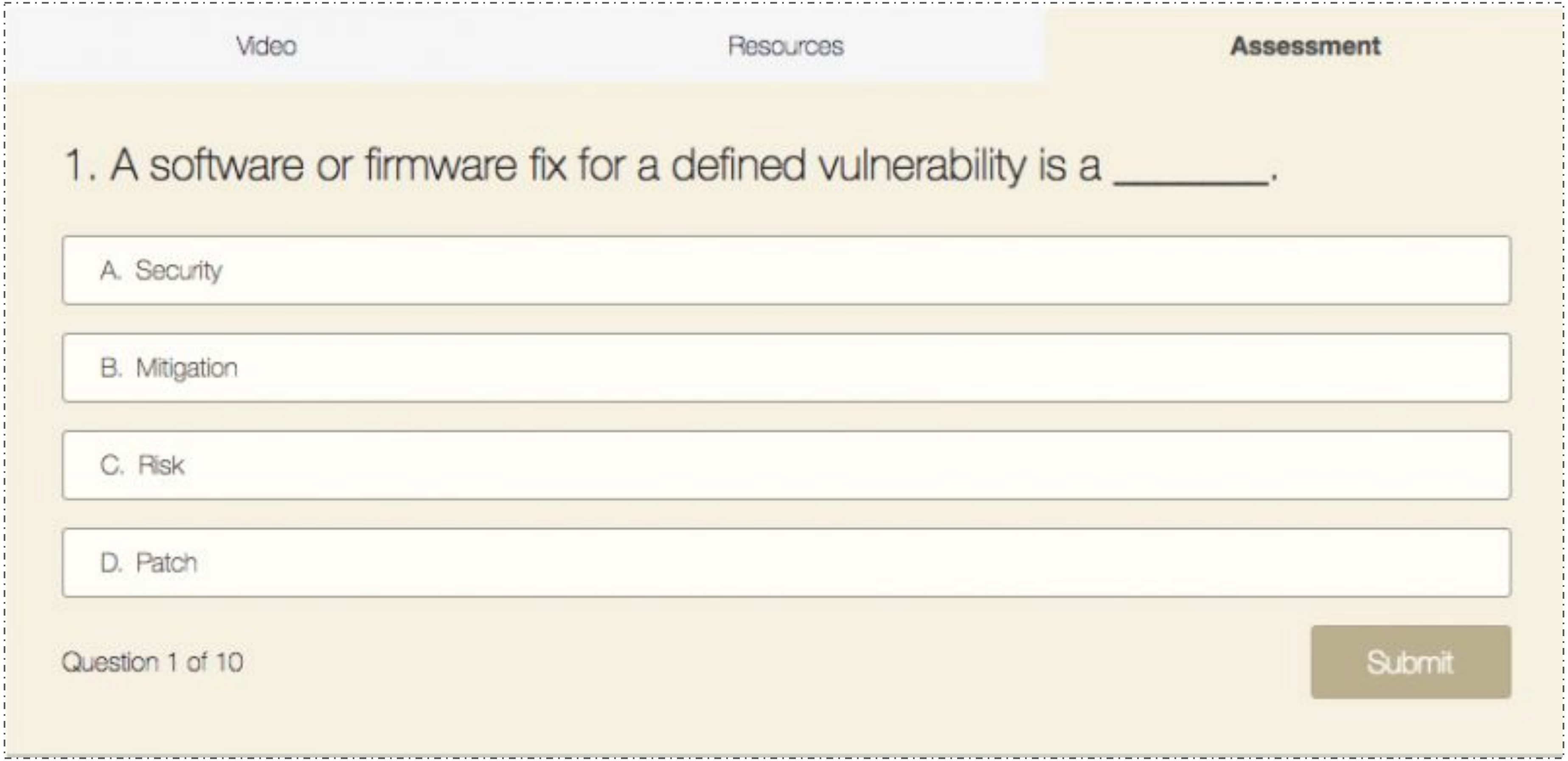
Changing Course on Education

ICAP Learning Framework

Engagement Activity	Example	Effectiveness
Passive	Watch a video	Worst
Active	Click through a tutorial	OK
Constructive	Answer an instructor’s questions	Better
Interactive	Solve a hands-on challenge	Best

https://files.eric.ed.gov/fulltext/EJ1044018.pdf

FACTS
Voice
00:51
PivotPoint
SECURITY



1. A software or firmware fix for a defined vulnerability is a _______.

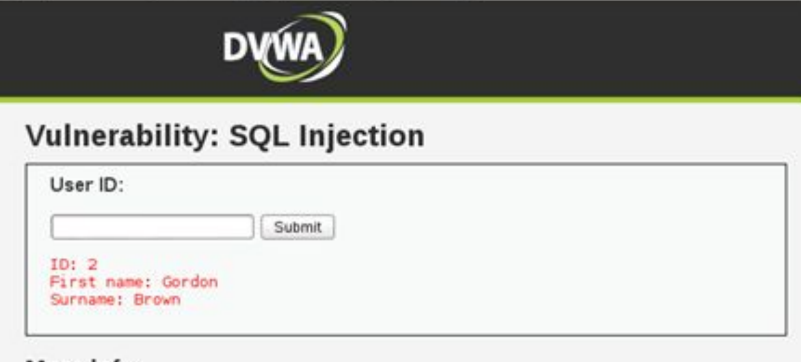
A. Security

B. Mitigation

C. Risk

D. Patch

Question 1 of 10

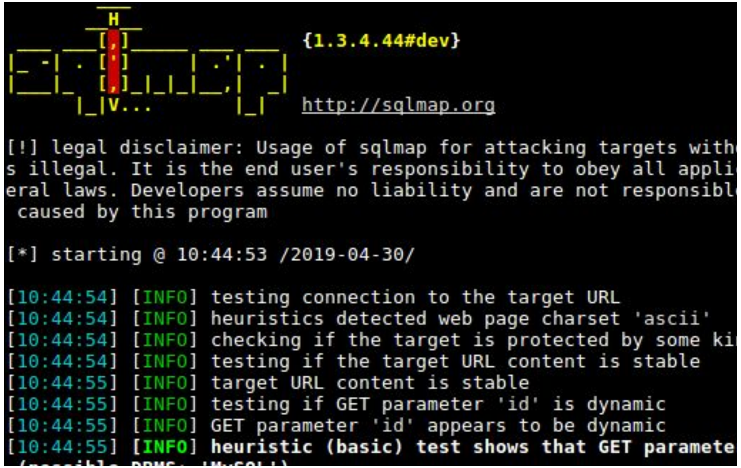


Vulnerability: SQL Injection

User ID:

Submit

ID: 2
First name: Gordon
Surname: Brown



```
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.3.4.44#dev}
|_ -| . [']     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets with
s illegal. It is the end user's responsibility to obey all appli
eral laws. Developers assume no liability and are not responsibl
 caused by this program

[*] starting @ 10:44:53 /2019-04-30/

[10:44:54] [INFO] testing connection to the target URL
[10:44:54] [INFO] heuristics detected web page charset 'ascii'
[10:44:54] [INFO] checking if the target is protected by some ki
[10:44:54] [INFO] testing if the target URL content is stable
[10:44:55] [INFO] target URL content is stable
[10:44:55] [INFO] testing if GET parameter 'id' is dynamic
[10:44:55] [INFO] GET parameter 'id' appears to be dynamic
[10:44:55] [INFO] heuristic (basic) test shows that GET paramete
 (possible DBMS: 'MySQL')
```

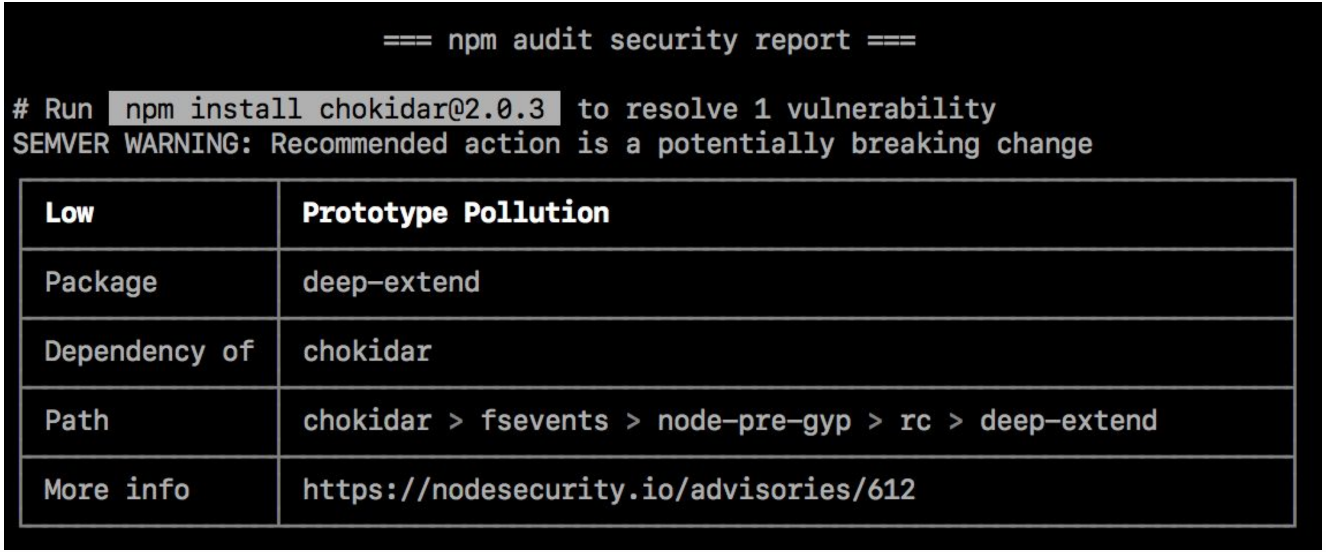
"It's the wrong approach. It's like going up to a parent and saying that their child is ugly and then expecting to have a conversation."

- Martin Knobloch, OWASP Chairman

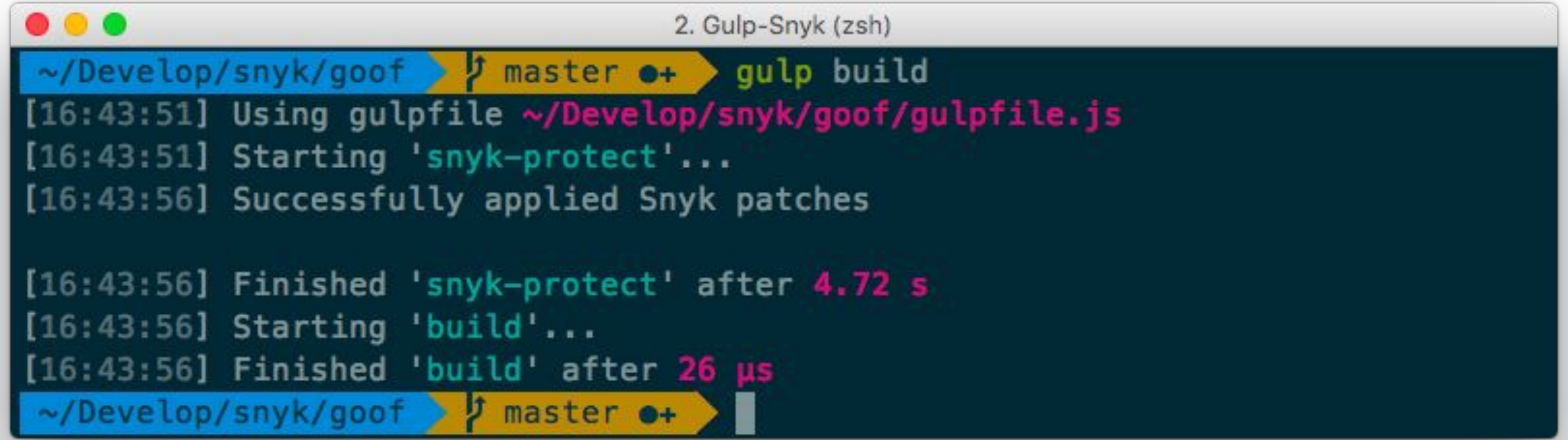
Explain engineering topics in engineering terms; speak to them as peers.

Don’t just tell developers that they can't be trusted to write secure code!

https://www.theregister.co.uk/2018/07/07/owasp_chairman_interview/

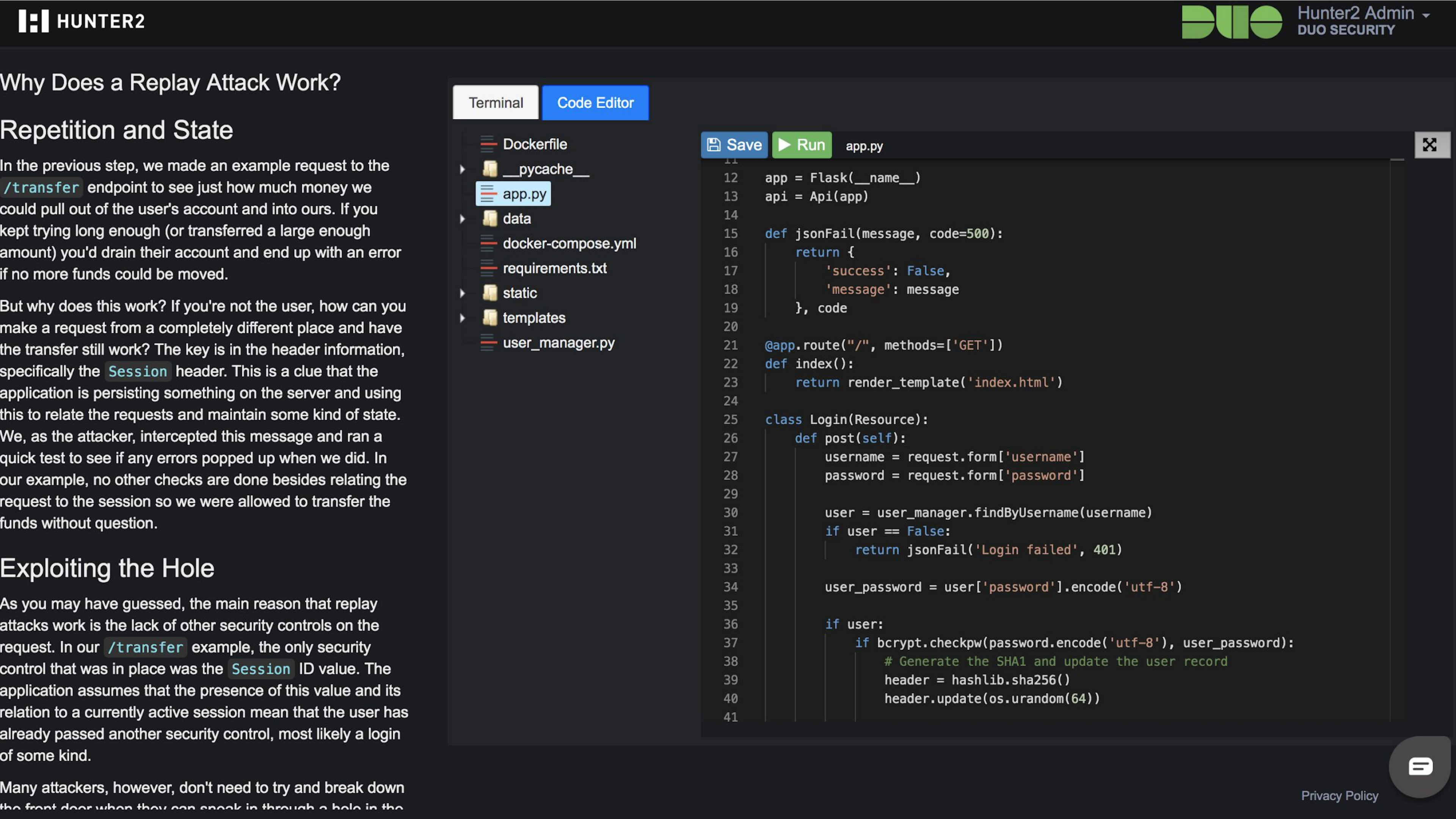


```
                    === npm audit security report ===

# Run  npm install chokidar@2.0.3  to resolve 1 vulnerability
SEMVER WARNING: Recommended action is a potentially breaking change
┌───────────────┬──────────────────────────────────────────────────────────────┐
│ Low           │ Prototype Pollution                                          │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Package       │ deep-extend                                                  │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Dependency of │ chokidar                                                     │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ Path          │ chokidar > fsevents > node-pre-gyp > rc > deep-extend        │
├───────────────┼──────────────────────────────────────────────────────────────┤
│ More info     │ https://nodesecurity.io/advisories/612                       │
└───────────────┴──────────────────────────────────────────────────────────────┘
```



2. Gulp-Snyk (zsh)

```
~/Develop/snyk/goof  master ●+  gulp build
[16:43:51] Using gulpfile ~/Develop/snyk/goof/gulpfile.js
[16:43:51] Starting 'snyk-protect'...
[16:43:56] Successfully applied Snyk patches

[16:43:56] Finished 'snyk-protect' after 4.72 s
[16:43:56] Starting 'build'...
[16:43:56] Finished 'build' after 26 μs
~/Develop/snyk/goof  master ●+ 
```



Why Does a Replay Attack Work?

Repetition and State

In the previous step, we made an example request to the `/transfer` endpoint to see just how much money we could pull out of the user's account and into ours. If you kept trying long enough (or transferred a large enough amount) you'd drain their account and end up with an error if no more funds could be moved.

But why does this work? If you're not the user, how can you make a request from a completely different place and have the transfer still work? The key is in the header information, specifically the `Session` header. This is a clue that the application is persisting something on the server and using this to relate the requests and maintain some kind of state. We, as the attacker, intercepted this message and ran a quick test to see if any errors popped up when we did. In our example, no other checks are done besides relating the request to the session so we were allowed to transfer the funds without question.

Exploiting the Hole

As you may have guessed, the main reason that replay attacks work is the lack of other security controls on the request. In our `/transfer` example, the only security control that was in place was the `Session` ID value. The application assumes that the presence of this value and its relation to a currently active session mean that the user has already passed another security control, most likely a login of some kind.

Many attackers, however, don't need to try and break down the front door when they can sneak in through a hole in the

```python
app = Flask(__name__)
api = Api(app)

def jsonFail(message, code=500):
    return {
        'success': False,
        'message': message
    }, code

@app.route("/", methods=['GET'])
def index():
    return render_template('index.html')

class Login(Resource):
    def post(self):
        username = request.form['username']
        password = request.form['password']

        user = user_manager.findByUsername(username)
        if user == False:
            return jsonFail('Login failed', 401)

        user_password = user['password'].encode('utf-8')

        if user:
            if bcrypt.checkpw(password.encode('utf-8'), user_password):
                # Generate the SHA1 and update the user record
                header = hashlib.sha256()
                header.update(os.urandom(64))

```

Growing the Community

Cyber Security Awareness Month - October 2019

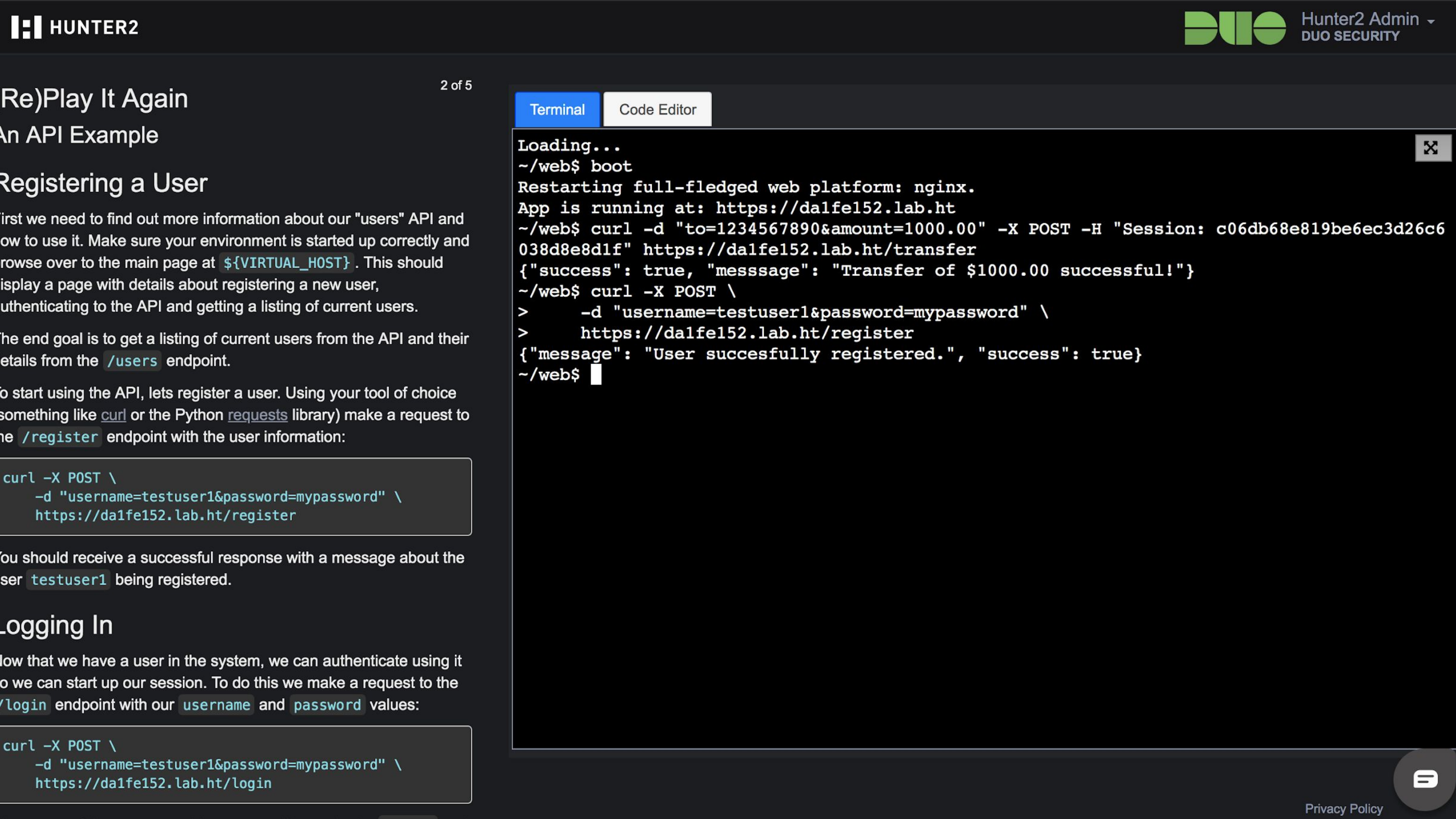
Introduction to Application Security

Advanced Application Security

- Utilizes a total of ~20 Hunter2 modules across courses
- Each course is designed to enable a day of training
- Speaker notes, lab guides, and other resources provided

Duo-created Lessons for Hunter2:

- Signing JSON Web Tokens
- HTTP Header Injection
- Replay Attacks
- Mass Assignment
- Securing Cookies
- Safe JSON Parsing



Re)Play It Again

An API Example

Registering a User

irst we need to find out more information about our "users" API and ow to use it. Make sure your environment is started up correctly and rowse over to the main page at `${VIRTUAL_HOST}`. This should isplay a page with details about registering a new user, uthenticating to the API and getting a listing of current users.

he end goal is to get a listing of current users from the API and their etails from the `/users` endpoint.

o start using the API, lets register a user. Using your tool of choice something like curl or the Python requests library) make a request to he `/register` endpoint with the user information:

```
curl -X POST \
    -d "username=testuser1&password=mypassword" \
    https://da1fe152.lab.ht/register
```

ou should receive a successful response with a message about the ser `testuser1` being registered.

Logging In

low that we have a user in the system, we can authenticate using it o we can start up our session. To do this we make a request to the `/login` endpoint with our `username` and `password` values:

```
curl -X POST \
    -d "username=testuser1&password=mypassword" \
    https://da1fe152.lab.ht/login
```

```
Loading...
~/web$ boot
Restarting full-fledged web platform: nginx.
App is running at: https://da1fe152.lab.ht
~/web$ curl -d "to=1234567890&amount=1000.00" -X POST -H "Session: c06db68e819be6ec3d26c6
038d8e8d1f" https://da1fe152.lab.ht/transfer
{"success": true, "messsage": "Transfer of $1000.00 successful!"}
~/web$ curl -X POST \
>     -d "username=testuser1&password=mypassword" \
>     https://da1fe152.lab.ht/register
{"message": "User succesfully registered.", "success": true}
~/web$ 
```

Join Us!

Reduce time-to-education by sharing newly identified risks and security best practices with the community

- Use community-driven labs for free training
- Contribute your own examples

hunter2.com/community

Shifting Knowledge Left

Keeping Up With Modern Application Security

Mark Stanislav

mstanislav@duo.com

DUO

Fletcher Heisler

fletcher@hunter2.com

HUNTER2

Join us! hunter2.com/community