RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: ASEC-W12

REPRONOW-SAVE TIME REPRODUCING AND TRIAGING SECURITY BUGS

Vinayendra Nataraja

Senior Product Security Engineer
Salesforce
@vinayendra

Lakshmi Sudheer

Security Researcher
Adobe, Inc.
@Lak5hmi5udheer

About Us

Vinayendra Nataraja

- Senior Product Security Engineer @ Salesforce
- Heads one of the largest private Bug Bounty program
- Twitter: @vinayendra

About Us

Lakshmi Sudheer

- Security Researcher @ Adobe
- Twitter: @Lak5hmi5udheer

Adobe

Agenda

- Bug Bounty Program
- The Triaging Process
- How ReproNow can help?
- How does the tool work?
- DEMO
- Future Plans

Bug Bounty Program

Bug Bounty (aka: Crowdsourced Security)

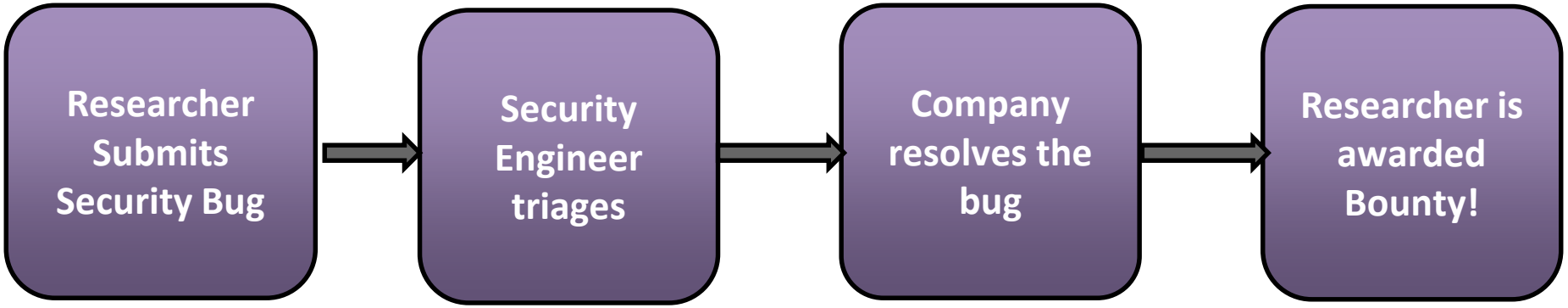
- Companies pay Hackers for a responsible disclosure
- Bounty is paid only for finding a valid security bug
- Company gets good security bugs for a fraction of the cost of consulting firms or hiring full time hackers
- Hackers get bragging rights / reputation + part time / full time income

Bug Bounty Program

Bug Bounty Setup

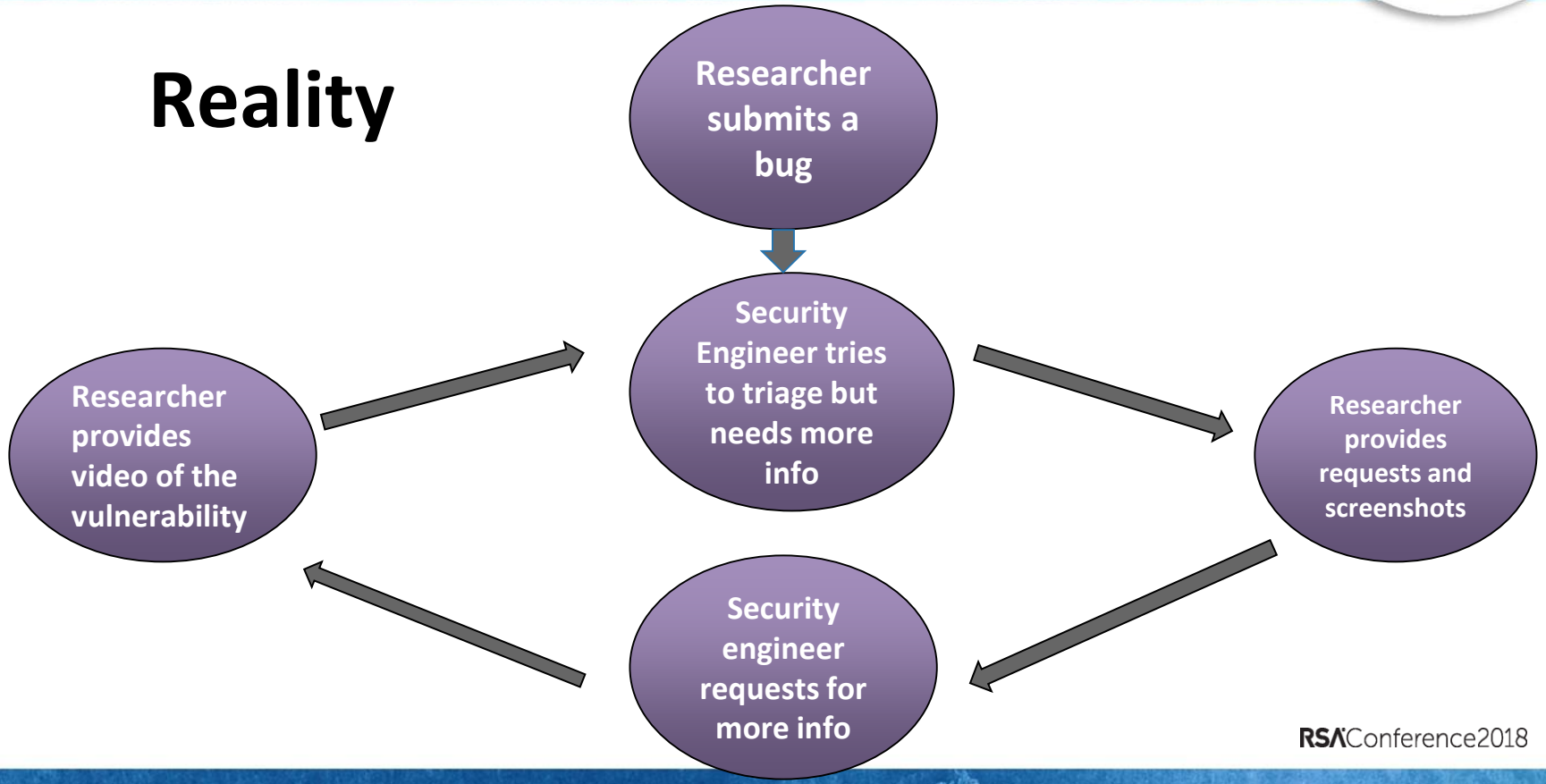
- Company decides on a structured program with defined scope and rules
- Hackers are setup with credentials to login and test
- Hackers submit bugs which need to be validated and fixed
- Company pays bounty to hackers for valid bugs

Expectation



Expectation Vs Reality

Reality



The Triaging Process

Information required for the Security Engineer to Triage

- Product
- URL
- Impact & Description
- Steps to Reproduce
- Request/Response
- Screenshots
- Video
- Documentation Links

Sometimes even more...

The Triaging Process : Report 1

[redacted] submitted a report to [redacted]. Jul 20th (2 months ago)

Description:

Attacker can update the user's Ad Frequency % using flash + 307 redirect trick by making post request to particular endpoint.

Step To Reproduce:

- Get logged at: https://cp-ng.pinion.gg
- Visit: http://geekboy.ninja/poc/freq.swf
- Ad Frequency should be updated.

[redacted] changed the status to **Needs more info**. Jul 27th (2 months ago)

We have problems reproducing this. Maybe it was fixed with another report about permissions - can you confirm the issue is gone?

[redacted] changed the status to **New**. Updated Sep 5th (5 days ago)

Hey its still working for me, maybe it didn't coz you tried to do with my account id [redacted]

[redacted] changed the status to **Needs more info**. Aug 1st (about 1 month ago)

We have problems reproducing - could you provide a poc video?

1 attachment:
F208974: 2017-07-21_07-12-52.mp4

[redacted] changed the status to **Triaged**. Aug 4th (about 1 month ago)

The Triaging Process : Report 2

dhaval submitted a report to Shopify. Oct 26th (11 months ago)

Hey

There seems to be a weird misconfiguration which leads to bypass of two factor authorisation

clayton closed the report and changed the status to **Not Applicable**. Oct 26th (11 months ago)

Thank you for your report.

This authentication mechanism is working as intended. If you switch your shop to use Google Apps as your login service, then authentication is handled through your Google account, and you would need to configure your Google account to require two-factor authentication.

dhaval posted a comment. Updated Oct 27th (11 months ago)

Hey @clayton @shopify

It's very disappointing when program does not read the full report and closes it as **N/A** and then ignores the comments

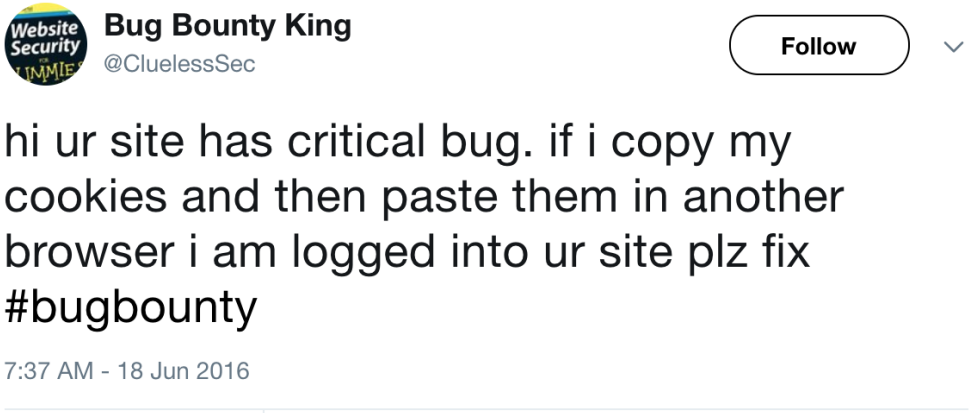
Here's my last attempt on describing the bug report while expecting a decent reply

Proof of Concept

clayton changed the status to **Triaged**. Nov 1st (10 months ago)

Thanks again for your report, and sorry for the confusion. Our engineering team is investigating the issue.

The Triaging Process



The Triaging Process

Triage Issues

- **Communication** Issues
  - **Non native English** speakers
  - Knowledge gap of product
  - Not everyone can write a **good report**
- **Complex** workflows
- **Long reports** will be time consuming to setup and triage
- WannaBe Hackers

The Triaging Process

Bug Bounty Pains for Companies

- **Time** spent
  - Going back and forth with researchers
  - Reproducing the bug all over again
- Reduced **efficiency**
- Increase in average **resolution time**
- Researcher Relationship
- Researcher Retention

The Triaging Process

"Triage takes time. So much time .... we paid people to help us run our bounty programs and it still took up a ton of time. Expect the time commitment to be 2–3 people full-time … The price you pay for the good issues is the cost of filtering out the junk."

- Collin Greene, Facebook

ReproNow: Introduction

What is ReproNow?

- A browser extension to capture **Desktop** and **Network** Traffic
- Has a responsive **UI** for Security Engineers to view and search
- **Hides** the traffic inside a mkv/webm video files
- Works **cross browser** using extensibility API, currently supports and

ReproNow: Introduction

What is ReproNow? (Continued)

- No server interaction, everything on **client**
- Export requests as **cURL**
- Store **history** in Local Storage
- Multiple options to capture Network
- **Open Source**

ReproNow: Working

How it Works?

- **getUserMedia API** to Capture Screen
- **WebRequest API** to Capture Networks
- Local storage to store the video and traffic
- **Mkv** files to hide the Network data.

ReproNow:Screen Capture

How Screen Capture Works?

- **chrome.desktopCapture API** allows to capture user screen.
- The desktopCapture API uses **getUserMedia API** which is a HTML5 API to capture camera/mic and also screen.
- getUserMedia API uses **MediaRecorder API** to convert recorded video into ArrayBuffer

ReproNow:Screen Capture

How Screen Capture Works? (Contd..)

- MediaRecorder Supports multiple **mimetypes** (we use webm,codec: VP8)
- The generated ArrayBuffer can we converted in blob URL using **createObjectURL API**
- The object can we piped into **<video>** tag for viewing or can we downloaded to user's device.

- video/webm
- video/webm;codecs=vp8
- video/webm;codecs=vp9
- video/webm;codecs=vp8.0
- video/webm;codecs=vp9.0
- video/webm;codecs=h264
- video/webm;codecs=H264
- video/webm;codecs=avc1
- video/webm;codecs=vp8,opus
- video/WEBM;codecs=VP8,OPUS
- video/webm;codecs=vp9,opus
- video/webm;codecs=vp8,vp9,opus
- video/webm;codecs=h264,opus
- video/webm;codecs=h264,vp9,opus
- video/x-matroska;codecs=avc1
- audio/webm
- audio/webm;codecs=opus

ReproNow: Screen Capture of Code


```
pending_request_id = chrome.desktopCapture.chooseDesktopMedia(
  SelectedDesktopOption, onAccessApproved);
```

```
navigator.webkitGetUserMedia({
  audio: audioConstraint,
  video: {
    mandatory: {
      chromeMediaSource: 'desktop',
      chromeMediaSourceId: id,
      maxWidth:screen.width,
      maxHeight:screen.height} }
}, gotStream, getUserMediaError);
```

ReproNow: Screen Capture of Code


```
function gotStream(stream) {

  window.stream = stream;
```

```
mediaRecorder = new MediaRecorder(window.stream, options);
mediaRecorder.onstop = handleStop;
mediaRecorder.ondataavailable = handleDataAvailable;
mediaRecorder.start(10); // collect 10ms of data
```

```
function handleDataAvailable(event) {
  if (event.data && event.data.size > 0) {
    recordedBlobs.push(event.data);
  }
}
```

ReproNow: Network Capture

How Network Capture Works?

- Two possible ways to intercept Network
  - chrome.debugger **API** which lets you attach to tabs and intercept traffic
    - But puts all tabs/windows into debugger mode :(
  - **chrome.webRequest API** observe and analyze traffic and to intercept, block, or modify requests in-flight
    - Cannot fetch response body :(

chrome.webRequest API - Request

chrome.webRequest API

onBeforeRequest

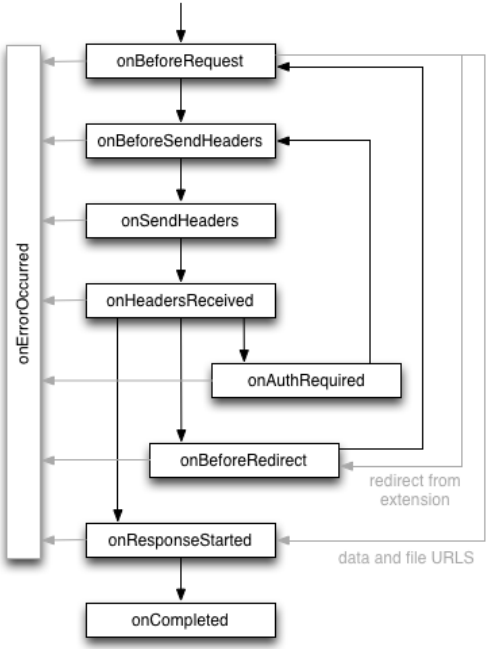
- Fires when a request is about to occur

onBeforeSendHeaders

- Initial headers have been prepared

onSendHeaders

- All headers are prepared (Read Only)



chrome.webRequest API - Response

onHeadersReceived

- Response header is received

onAuthRequired

- Request requires authentication of the user.

onBeforeRedirect

- Redirect is about to be executed

onResponseStarted

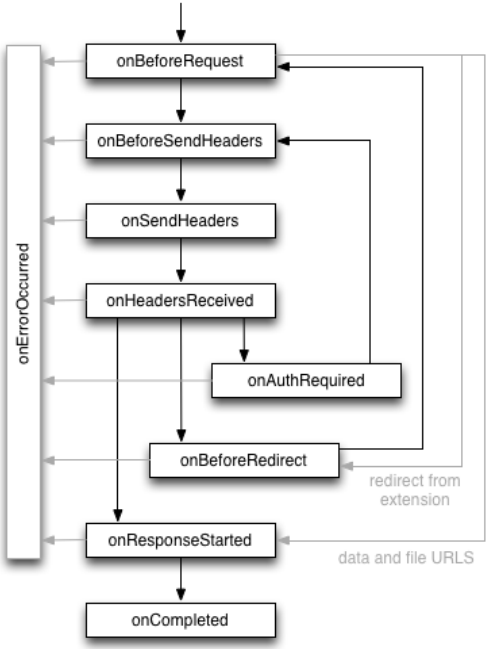
- First byte of the response body is received (Read Only)

onCompleted

- Request was succesful

onErrorOccurred

- Request Failed



chrome.webRequest API (contd)

ReproNow uses

- **onBeforeRequest** to get URL, method and Request Body
- **onBeforeSendHeaders** to get all request headers
- **onHeadersReceived** to get all response headers

Network Capture (Cont'd)

- Start clock between Screen and network to keep both in sync
- Option provided for user to capture network when navigating between tabs
  - Capture a specific tab
  - Capture all tabs
  - Capture network of all activated tabs
  - Capture traffic only activated tab

Network Capture: Code


```
chrome.webRequest.onBeforeRequest.addListener(addWebReq,
{
    urls:["<all_urls>"],
    tabId:tabid,
    types:["main_frame","sub_frame","xmlhttprequest"]
}
,
['requestBody']
);
```

```
function addWebReq(details)
{
    if(details.requestBody)
        req.get(details.requestId).requestBody=details.requestBody;
    if(details.requestHeaders)
        req.get(details.requestId).requestHeaders=details.requestHeaders;
    if(details.responseHeaders)
        req.get(details.requestId).responseHeaders=details.responseHeaders;
}
```

Exporting Screen and Network

How do we share **Screen + Network** ?

Requirement:

All video operations must happen **Client-side**

Solution:

- Downloading 2 files - Network and video separately
- Zipping network file and the video file

What if we can insert the network file inside video?

Inserting JSON into Video file

The Need:

1. Video Format that support adding a JSON file without breaking the video.
2. API/Tool to perform client side operation to manipulate the video file.

Inserting JSON into Video file

The Need:

1. **Video Format that support adding a JSON file without breaking the video.**
2. API/Tool to perform client side operation to manipulate the video file.

Video Formats

Video formats

Most common formats:

- **MP4**
- **MKV**
- AVI
- FLV

MPEG-4

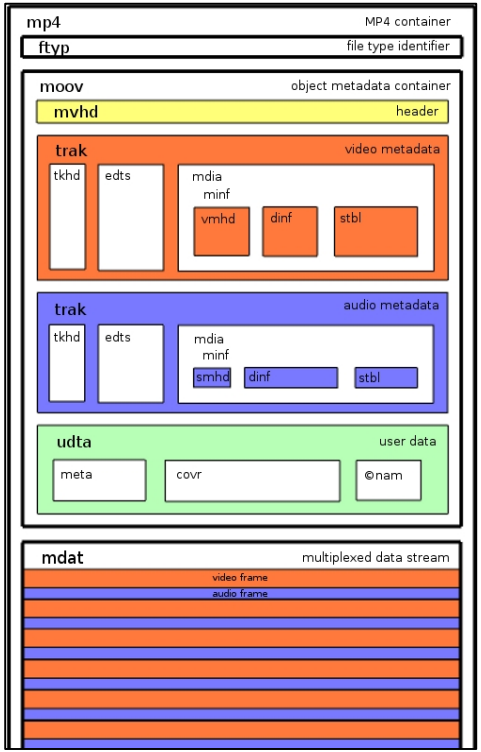
MPEG-4 (MP4)

- Digital multimedia format **container**
- Directly based on from **QuickTime** File Format
- Stores audio, video, subtitles and images
- File Extensions:
  - Video + Audio : **.mp4**
  - Audio only : **.m4a**
  - Raw MPEG-4 visual bitstream : **.m4v**

MPEG-4

MPEG-4 Structure

- **ftyp** - file type
- **moov** - contains metadata (song title, author, url, and other information)
- **mdat** - contains the audio frames and video frames



MKV

Matroska(MKV)

- **Open source** container
- **Free container** format
- Holds **unlimited** audio tracks, video tracks, subtitles, files
- **WebM** was specifically designed for the internet is derived from Matroska and can be used interchangeably
- File Extensions:
  - Video + Audio : .mkv
  - Stereoscopic Video: .mk3d
  - Audio only : .mka

MKV: Structure

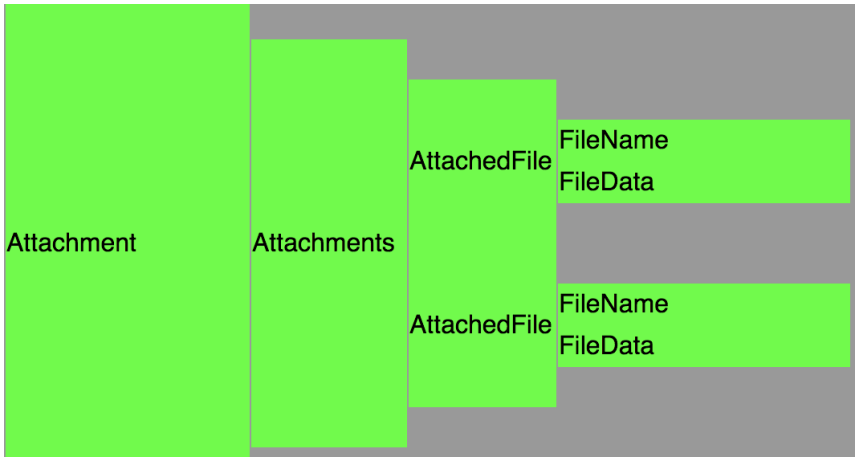
MKV Container

- **Header:** EBML version
- **Meta Seek Information:** Index to other groups like Track information, Chapters, Tags, Cues, Attachments
- **Segment Information:** Basic information, title, UNique ID
- **Track section:** Basic information about each of the tracks, codec, Sample Rate

MKV: Structure

- **Chapters:** Section lists all of the Chapters
- **Clusters:** contain all of the video frames and audio for each track
- **Cueing Data:** Cues are the index for each of the tracks. This is used for seeking to a specific time when playing back the file
- **Attachment section:** Allows attaching any file type to the video.
- **Tagging section:** This contains Tags that relate to the the file and tracks like the actors, singers and song information

Attachment Section



Why Mkv?

- **No restrictions** on filetype in the attachment section
- **Webm** is built on MKV providing greater flexibility on the client side
- Almost all **browsers** support MKV/WebM
- Easy to store and dump a JSON file

Inserting JSON into a Video file

The Need:

1. Video Format that support adding a JSON file without breaking the video.
2. **API/Tool to perform client side operation to manipulate the video file.**

Ffmpeg

Fast Forward MPEG (ffmpeg)

- Free **Multimedia Framework**
- **Encodes, Decodes, Muxes, Demuxes** audio and performs other operations on video and audio files
- Contains libraries like **libavcodec** which supports more than 200 video/audio formats
- One of the most widely used platform for audio/video manipulation

Embedding JSON

What we used..

- **ffmpeg.js** - Ffmpeg that can run on a browser

  Ffmpeg

  -i **video.mkv**
  -attach **network_data.json**
  -metadata:s:t mimetype=application/json

  output.mkv

- **ts-ebml** - Node.js library that Encodes/Decodes Mkv files on browser using Browserify

Putting it all together

1. **Record the Screen** using getUserMedia API and store it as arrayBuffer
2. **Capture Network** using WebRequest API and store it as JSON
3. **Save** the video and JSON in localstorage
4. **Preview** the video using <video> tag
5. **Sync** Network with Screen
6. **Download** the Video in MKV by attaching the network JSON as an attachment

ReproNow: Demo

DEMO

How Can ReproNow Help?

For Bug Bounty hunters

- Awesome **screen capture tool**!
- **Preview** the capture before sharing
- Ability to copy paste **raw** requests/responses and generate **cURL** script for specific ones
- A **video** (with network) speaks thousand words
- **No server** is involved and is also **open source**, therefore no need to trust a vendor to keep your data secure
- Faster Triage = Faster **Bounty**

How Can ReproNow Help?

For Organizations

- Reproduce without **manually** going through the steps
- Helps in reducing **noise** in your bug bounty program
- Saves your Security Engineer's **Time**
- Saves **money** by reducing expenditure on managed services
- Can be used for **QA** internally
- Faster Traige = Faster Bounty = **Researcher Retention**

Extending ReproNow

Extending ReproNow

- **Automatically Triage** Bug Bounty Bugs
  - Replay the network requests in internal builds to verify the bug
  - Replay the network in various browsers
- Help **Detect login** to run ZAP
  - Salesforce Appexchange Security Review uses Chimera (Appsec USA 2015 - Tim Bach)
  - Chimera login detection can be improved using ReproNow

Links

REPRONOW

Github: repro-now.com/github

Chrome Extension: repro-now.com/chrome

Website: repro-now.com

Thank you!

repro-now.com

@vinayendra

@Lak5hmi5udheer